

Contents

1	Introducción	1
2	Capa de enlace	35
3	IPv4	66
4	IPv6	90
5	UDP y TCP	110
6	DNS y Firewall	138

1 Introducción

Teoría

Hoja 1

Introducción

Actualmente, el concepto de un "centro de cómputo" donde una computadora principal se ocupa del cálculo y procesamiento de información ha quedado obsoleto. Este modelo fue reemplazado por otro en el que un gran número de computadoras con procesamiento propio intercambian entre sí para hacer el trabajo.

Diremos que una red de computadoras es un conjunto de computadoras autónomas interconectadas que acceden a la información concentrada en un servidor de archivos principal o distribuida en más de ellas.

Las redes de computadoras minimizan los problemas de distancia y comunicación, permitiendo el acceso a la información desde cualquier punto de la red. Algunos usos son:

• Compartir programas y archivos: hay muchos paquetes de software que permiten centralizar la información (ej.: DBMS). Los usuarios pueden almacenar sus archivos en el servidor para que sean accesibles por otros usuarios. La finalidad de un servidor de archivos y programas es evitar la redundancia de información en el sistema.

• Compartir recursos de red: entre los recursos tenemos impresoras, digitalizadores, dispositivos de almacenamiento, etc. Su relación costo/beneficio es baja al encontrarse disponibles en la red.

• Correo electrónico: permite la comunicación interna de los usuarios del sistema facilitando cooperación entre personas alejadas geográficamente, o control de tareas asignadas en una empresa.

• Gestión centralizada: debido a que la mayoría de los recursos se encuentran organizados alrededor de un servidor, su gestión resulta más fácil. Las copias de seguridad, organización y optimización del sistema se pueden llevar a cabo desde un único lugar.

• Aplicaciones dinámicas, acceso a información remota, comunicación de persona a persona, entretenimiento interactivo, comercio electrónico, etc.

• Usuarios móviles: las redes inalámbricas otorgan al usuario la posibilidad de mantenerse conectado sin elementos físicos, permitiéndole realizar sus actividades desde cualquier punto.

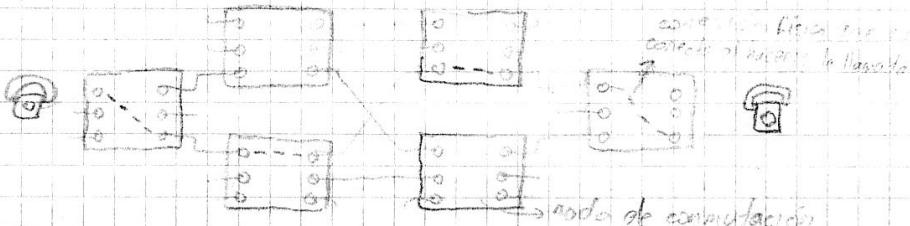
Tecnología y clasificación de redes

Tenemos varios criterios para clasificar las redes:

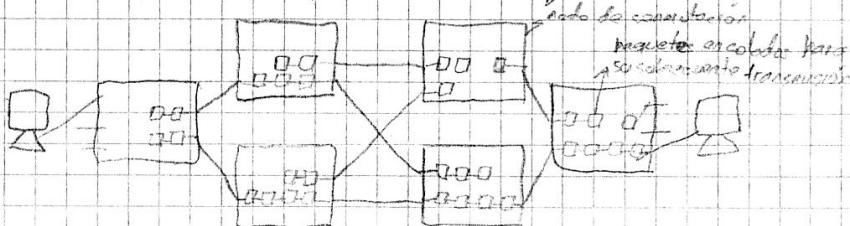
Clasificación

- Por tipo de conmutación:

- Circuitos conmutados: el equipo de conmutación hace una trayectoria física de "cobre" que vaya desde el origen al destino. Esta técnica era utilizada en la telefonía antigua.



- Conmutación de paquetes: los mensajes que se transmiten desde el origen al destino se fraccionan en paquetes y se envían al primer nodo de conmutación. Éste los inspecciona y retransmite vía troncales a los nodos de conmutación apropiados hasta alcanzar su destino.



- Según el tipo de transmisión:

- Redes de difusión: todos las máquinas comparten un único canal de comunicación. En éste circulan los mensajes, también llamados paquetes, que pueden ser enviados por cualquier máquina conectada. Estos son recibidos por las otras máquinas y lo procesan si son los destinatarios del mismo (verificando un campo de dirección del paquete). Por lo general, los sistemas de difusión permiten el direccionamiento de un paquete a todos los destinos (broadcasting); y algunos también soportan la transmisión a un subconjunto de máquinas (multicasting).

- Redes punto a punto: consisten en muchas conexiones entre pares individuales de máquinas. Para que un paquete vaya del origen al destino debe pasar por uno o varios máquinas intermedias. Es posible que haya varias rutas con longitudes diferentes, de manera que es importante encontrar las más eficientes. La transmisión de punto a punto se conoce como unicasting.

- Según el alcance:

- Redes de área personal: están destinadas para una sola persona (ej: mouse a computadora).

- Redes de área local (LAN): son redes que cubren uno o varios edificios en un radio de hasta unos pocos kilómetros. Se usan ampliamente para conectar computadores personales y estaciones de trabajo en oficinas de una empresa y de fábricas para compartir recursos e intercambiar información.

Hoja 2

Debido a su tamaño, los tiempos de transmisión son bajos y escurridos de antemano; esto permite utilizar ciertos tipos de diseño y simplifica la administración de la red. Las LANs tradicionales se ejecutan a una velocidad de 10 a 100 Mbps, tienen un retraso bajo y cometen pocas errores. Algunas LANs más novedosas alcanzan velocidades de hasta 10 Gbps.

• Redes de área metropolitana (MAN): son básicamente una mayor versión de las LAN y en general su tecnología no difiere mucho. Su radio de acción es de alrededor 10 Km. Un ejemplo conocido es la red de televisión por cable disponible en una ciudad.

• Redes de área amplia (WAN): cubren una gran área geográfica (en país o en continental). Estas redes aplican la tecnología punto a punto, orientada a optimizar el aprovechamiento de los enlaces de enlace entre sus componentes. En general, las WAN son conjuntos de LANs interconectados punto a punto por roteadores.

• Internet: la red de redes, su cobertura se extiende a todo el mundo.

Según su tipología:

Nombre	Estructura	Enlaces	Tráfico	Vínculo	Costo	Añadir dispositivo
Anillo		n cables	Baja congestión	Fibra óptica	Bajo	Difícil
Bus		n cables	Baja congestión	Fibra óptica / Par trenzado	Bajo	Fácil
Malla		($\frac{n(n-1)}{2}$) cables	Relativo	Par trenzado Cable coaxial	Alto	Difícil
Estrella		N-1 cables	Alta congestión	Fibra óptica	Relativo	Fácil

Software de red

Los primeras redes de computadoras se diseñaron teniendo al hardware como punto principal, lo que dificultaba la conectividad con otras redes que fueran diferentes. En la actualidad, el software está altamente estructurado.

Jerarquías de protocolos

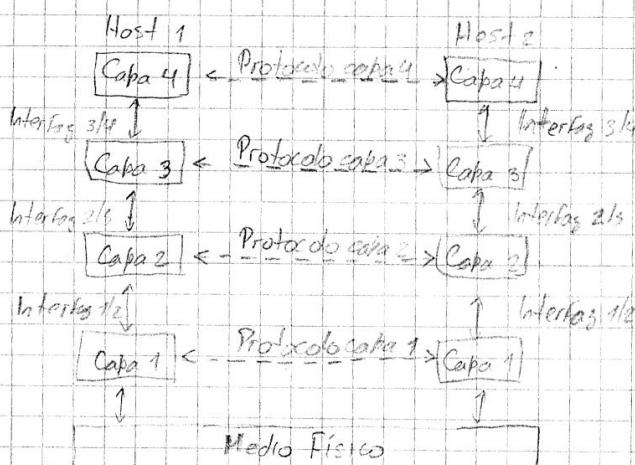
Para reducir la complejidad de su diseño, la mayoría de las redes están organizadas como una

Jararquía de Protocolos

pila de capas o niveles, cada una construida a partir de la que es la debajo. El número de capas, el nombre, su contenido y función difieren de red a red. El propósito de cada capa es ofrecer determinados servicios a las capas superiores (sin que éstas sepan cómo se implementaron). Esto concepto es conocido como encapsulamiento de datos.

La capa n de una máquina se comunica con la capa n de otra máquina mediante un protocolo de capa n . Básicamente, un protocolo es un acuerdo entre las partes en comunicación sobre como llevar a cabo. Las entidades que abarcan las capas análogas en diferentes máquinas se conocen como pares (peers).

En realidad, los datos no se transfieren directamente entre las capas análogas, sino que cada capa n pasa los datos y la información de control a la capa inmediatamente inferior, hasta alcanzar la capa física donde realmente se produce la comunicación.

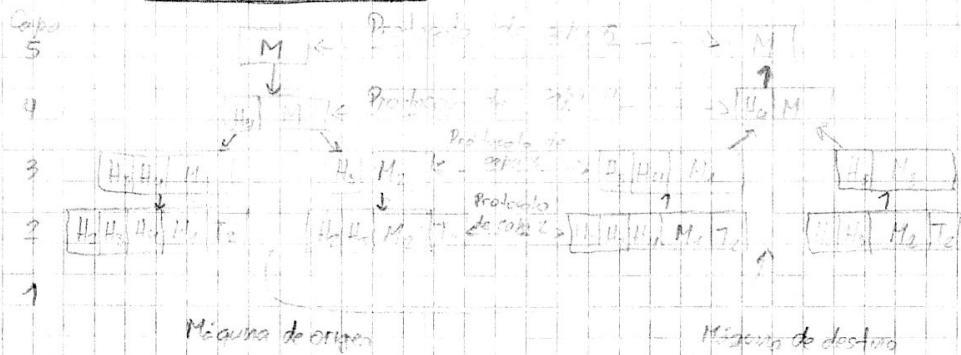


Los datos entre capas adyacentes se transfieren mediante una interfaz. Esta define qué operaciones y servicios primitivos tiene la capa más baja a disposición de la capa superior inmediata. Una consideración importante en el diseño de redes es definir interfaces limpias entre las capas, que desempeñen un conjunto específico de funciones bien entendidas. Además de minimizar la información que se debe pasar entre las capas, las interfaces bien definidas simplifican el reemplazo de la implementación de una capa con una implementación totalmente diferente, mientras ofrece exactamente el mismo conjunto de servicios.

El conjunto de capas y protocolos se denomina arquitectura de red. Su especificación debe contener información suficiente para permitir escribir programas o diseñar software de modo que se cumpla correctamente con el protocolo apropiado. Aunque las interfaces están ocultas en cada máquina, no necesitan ser rigidas mientras cumplen los protocolos. La lista de protocolos utilizados por un sistema se conoce como pila de

protocolos.

Comunicación entre capas



El Funcionamiento general es el siguiente:

- 1) La capa superior genera un mensaje (M) y se lo pasa a la capa inmediata inferior.
- 2) La capa siguiente toma el mensaje y le agrega un encabezado (H) que contiene información de control; opcionalmente podría agregar una cola (T). Podría suceder que la capa inferior tenga un límite para el tamaño del mensaje, entonces M se divide en dos mensajes: M₁ y M₂.
- 3) Este proceso sigue hasta la capa inferior, donde se transiten los paquetes por el medio físico hacia la otra máquina.
- 4) La capa inferior receptora toma los paquetes y los reúne utilizando el protocolo. Luego pasa el resultado a la capa inmediata superior, quitando encabezados y colas.
- 5) Continúa el proceso hasta llegar a la capa superior, obteniendo el mensaje (M).

La abstracción de procesos hace simplificar el diseño de una pila de protocolos puesto que éste puede fragmentarse en pequeños problemas de diseño de las capas individuales.

Generalidades en el diseño de capas

Debido a que una red está formada por varias computadoras, cada una corriendo varios procesos distintos, debe ser posible que un proceso determinado establezca conexión con otro proceso específico en otra máquina. Como consecuencia de los múltiples destinos es necesario implementar un mecanismo de direccionamiento.

Otro punto se refiere a la forma en que se transferirán los datos: simple (unidireccional), half duplex (bidireccional pero no a la vez) o full duplex (bidireccional). También se deben determinar los canales lógicos por conexión que están disponibles y la prioridad de cada uno.

muchos canales proporcionan un canal para datos normales y otro para datos urgentes).

El control de errores es un aspecto importante porque los circuitos de comunicación física no son perfectos. Ambos extremos deben usar el mismo código de detección y corrección de errores. Además, el receptor debe poder avisar qué mensajes se reciben correctamente y cuáles no.

No todos los canales de comunicación conservan el orden en que se envían los mensajes. El protocolo debe incluir un mecanismo que permita al receptor corregir posibles pérdidas de secuencia y reordenar paquetes para obtener el mensaje original. Una solución es numerar los paquetes.

Un aspecto a tener en cuenta es la existencia de distintos tipos de máquina dentro de una red, para evitar que un emisor rápido sobrepase la capacidad de procesamiento de un receptor lento. Debe buscarse un mecanismo que permita al emisor conocer la capacidad de procesamiento del receptor. Esto se conoce como control de flujo.

Otro problema es la longitud de los mensajes que se transmiten, lo que conduce a mecanismos para desensamblar, transmitir y reensamblar mensajes. Un aspecto relacionado es el problema de procesos que quieren transmitir datos en cantidades pequeñas que hacen inefficiente el envío por separado. Una solución es unir varios mensajes pequeños en uno grande (siempre que los pequeños tengan un destino común) y desmembrarlo cuando llegue a su destino.

Respecto a la optimización del canal se implementan mecanismos de multiplexación que permiten que dos o más procesos utilicen el mismo canal. Cuando hay múltiples rutas entre el origen y el destino, debe elegirse la mejor de ellas de acuerdo a la distancia y la carga de tráfico (rutinamiento).

Servicios orientados a conexión y sin conexión

Los cables pueden ofrecer dos tipos de servicios a los usuarios que están sobre ellos:

- El servicio orientado a la conexión se asemeja al sistema telefónico, el usuario primero establece una conexión, la utiliza y luego la abandona. El aspecto esencial es que funciona como un tubo: el emisor envía datos en un extremo y el receptor los toma en el otro (conservando el orden). En algunos casos, el emisor y el receptor efectúan una negociación sobre los parámetros de la conexión.

- Los servicios sin conexión se asemejan al sistema postal; cada mensaje lleva su dirección de destino y es enviado a través del sistema, independientemente de los demás. En general, el primer mensaje en enviarse es el primero en llegar, pero esto no siempre es así.

Los servicios pueden caracterizarse por su calidad de servicio. Algunos servicios son con-

Hoja 4

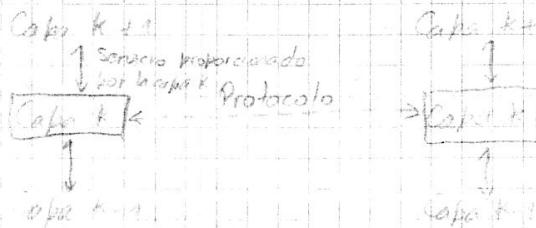
fiables en el sentido de que nunca pierden datos. Para esto, debe implementarse una forma de recibir confirmación de que el mensaje llegó al receptor. Esto induce sobrecargas y retardos, que con frecuencia son valiosos pero a veces indeseables.

Un ejemplo de servicio orientado a la conexión confiable es la transferencia de archivos, pero si no queremos los retrasos de confirmación tenemos el caso del tráfico de voz digitalizada. En otras ocasiones los servicios sin conexión requieren confirmación, como el correo certificado; pero si no es el caso, como en correo electrónico, hablamos de servicio de datagramas.

Otro servicio más es el de solicitud-respuesta, en el que el emisor transmite un datagrama con una solicitud y recibe otro datagrama con la respuesta.

Relación entre servicios y protocolos

Servicios y protocolos son conceptos distintos, aunque frecuentemente se confunden. Un servicio es un conjunto de primitivas (operaciones) disponibles para la capa inmediata superior, que son utilizadas por una entidad de la capa superior en forma transparente. Por otro lado, un protocolo es un conjunto de reglas que establecen el formato y el significado de los fragmentos o mensajes que se intercambian las entidades o pares de una capa determinada. Los entidades usan protocolos para implementar sus definiciones de servicio. Cada capa puede cambiar sus protocolos cuando desee, siempre y cuando no cambie el servicio visible a sus usuarios.



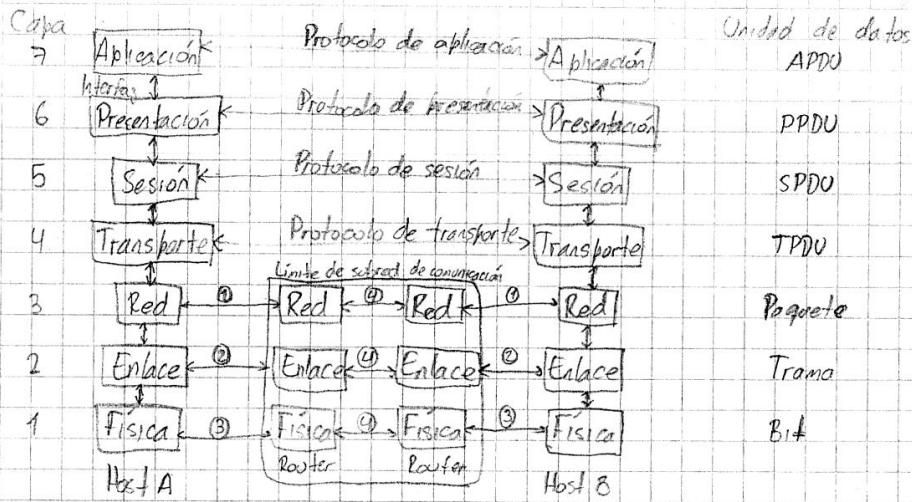
Modelos de referencia

Veremos dos arquitecturas de redes muy importantes: los modelos de referencia OSI y TCP/IP. Aunque los protocolos asociados con el modelo OSI ya casi no se usan, el modelo en sí es muy general y aún es válido, y las características tratadas en cada capa son muy importantes. El modelo TCP/IP tiene las propiedades opuestas: el modelo en sí no se utiliza mucho pero los protocolos sí.

El modelo de referencia OSI

International Standardization Organization

Este modelo está basado en una propuesta desarrollada por la ISO, como un primer paso hacia la estandarización internacional de los protocolos utilizados en varias capas. Sus siglas derivan de Open Systems Interconnection.



- ① Protocolo de router-host de la capa de red
- ② Protocolo de router-host de la capa de enlace
- ③ Protocolo de router-host de la capa física
- ④ Protocolo de subred interna

El modelo OSI fue creado según los siguientes principios:

- Cada capa se creó para satisfacer un nivel de abstracción distinto.
- Cada capa tiene una función específica y claramente definida.
- La función de cada capa se eligió teniendo en cuenta la función del modelo como estándar internacional.
- Los límites de cada capa se determinan con el objeto de minimizar el tráfico de información a través de las interfaces.
- El número de capas debe ser suficiente para no agrupar demasiadas funciones en cada una ya sea para facilitar la comprensión del funcionamiento del modelo. *Poco a la vez, no muchos capas para que no se vuelva inmanejable*
- Capa física

En esta capa se lleva a cabo la transmisión de bits puros a través de un canal de comunicación. Se tratan todos los aspectos físicos de la comunicación: medio de transmisión utilizado, temporización de señales, niveles de tensión, características mecánicas de los terminales de conexión, sincronismos y delimitadores a nivel de bit, cuando hay información en el medio, comienzo y fin de la comunicación,

implementación física de la comunicación duplex, etc.

Capa de enlace

La tarea principal de esta capa es transformar un medio de transmisión lento en una línea de comunicación rápida, al llegar a la capa de red, aparece libre de errores de transmisión. También se ocupa de la regulación del flujo de datos: para evitar que un transmisor rápido saturé a un receptor lento.

En algunas redes, generalmente las LAN, se divide esta capa en dos subcapas: LLC (Logical Link Control) y MAC (Medium Access Control).

Capa de red

Esta capa controla las operaciones de la subred y se encarga de encajar los paquetes de datos dentro de la subred (dinámicamente o estáticamente), controlar la congestión que se produce por exceso de paquetes, controlar el flujo entre nodos y anfitriones, posiblemente realizar trabajos de contabilidad y controlado del uso de la comunicación, dar un formato de direcciones para calcular el enrutamiento (direcciones universales y cínicas para cada terminal), asegurar interconexión de redes heterogéneas, etc.

En las LAN no existen problemas de encajamiento o congestión, por lo que la capa de red es a veces delegada, o no siguiendo existe.

Este es el último nivel que se implementa dentro de la subred (máquinas intermedias entre los hosts). Los protocolos superiores sólo reconocen la máquina de destino.

Capa de transporte

Es la primera capa que establece un diálogo extremo a extremo, es decir, no se utilizan máquinas intermedias como en las capas 1, 2 y 3.

La función básica de esta capa es aceptar los datos provenientes de las capas superiores, dividirlos en unidades más pequeñas si es necesario, pasar éstas a la capa de red y asegurarse de que todas las piezas lleguen correctamente al otro extremo. Además, esto debe hacerse de forma transparente y con trámites normalizados independientes de la capa de red.

La capa de transporte también determina qué tipo de servicio proporcionar a la capa de

sesión. El tipo de conexión de transporte más popular es un canal punto a punto libre de errores que respeta el orden de envío. Sin embargo, otros tipos de servicio de transporte posibles son el envío de mensajes caóticos (no garantiza orden) y la difusión a múltiples destinos.

En servicios orientados a conexión, la capa de transporte debe encargarse de solicitar una conexión a la red y liberarla cuando termine la conexión, puede multiplicar varias conexiones de sesión en una de transporte para reducir gastos y también puede dividir una conexión de sesión en varias conexiones de red para aumentar el caudal de datos. En un servicio sin conexión, debe otorgar los PDUs para entregárselos a la capa de sesión y tiene que establecer un control de flujo entre las dos terminales.

Capa de sesión

Esta capa permite que los usuarios de máquinas diferentes establezcan sesiones entre ellos. Las sesiones ofrecen varios servicios como el control del diálogo, la administración del token y la sincronización (agrega checkpoints para permitir recuperar la sesión luego de una caída).

El diálogo entre dos máquinas puede ser dúplex o half-dúplex.

Capa de presentación

A esta capa le corresponde la sintaxis y la semántica de la información transmitida. Permite que dos computadoras con distintos formatos de representación de datos se puedan comunicar a través del uso de una sintaxis abstracta, junto con una codificación estática.

Esta capa también proporciona cifrado de datos para garantizar su seguridad, aunque no es exclusivo de la capa. También puede implementar compresión de los datos.

Capa de aplicación

Proporciona una serie de servicios que un usuario utiliza con frecuencia, como correo electrónico, transferencia de archivos, llamadas a procedimientos remotos, directorios, consultas a bases de datos, etc. Para esto contiene varios protocolos muy comunes (ej: HTTP).

Criticas al modelo OSI

- Aparición no portuna: cuando se crearon los protocolos OSI, los protocolos TCP/IP competidores ya eran ampliamente utilizados por universidades investigadoras. Luego, ninguna empresa dio el primer paso para utilizar OSI, y éste nunca prosperó.

- Mala tecnología: tanto el modelo como el protocolo tienen defectos. La elección de los 7

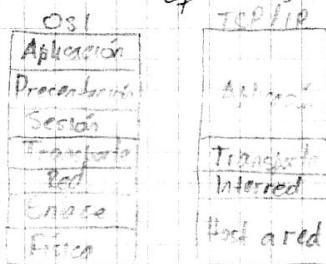
capas fue más política que técnica; por esto hoy dos capas casi vacías (sesión y presentación), y dos que están saturadas (enlace y red). Además, el modelo OSI es extraordinariamente complejo; sus estándares son difíciles de implementar y de operación inefficiente. Además, algunas funciones (como direccionamiento, control de flujo y control de errores) aparecen en varias capas.

- Malas implementaciones: ante la enorme complejidad del modelo y los protocolos, las implementaciones iniciales eran grandes, pesadas y lentas. Se asoció OSI con "baja calidad" y esta imagen persistió a pesar de la mejora en los productos. Por el contrario, una de las primeras implementaciones de TCP/IP era parte de UNIX y era bastante buena (además de gratis). Se produjo una espiral ascendente de calidad y uso en este caso.

- Malas políticas: se tenía la idea de que OSI era el producto de un manjo de burocratas gubernamentales intentando poner en marcha un estándar técnicamente inferior al mundo de investigadores y programadores pobres que intentaban realmente desarrollar redes. Esto no ayudó en nada a la popularidad de OSI.

El modelo de referencia TCP/IP

La arquitectura TCP/IP está muy difundida debido a la expansión de Internet. En el momento de su creación, se buscaba la capacidad para conectar múltiples redes de una manera sólida. Otro objetivo fue que la red pudiera sobrevivir a la pérdida de hardware de la subred sin que las conversaciones existentes se interrumpieran. Además, se necesitaba una arquitectura flexible debido a que se preveían aplicaciones con requerimientos divergentes.



Capa de host a red (Interfaz de red)

El modelo de referencia TCP/IP en realidad no dice mucho acerca de lo que pasa aquí, sólo puntualiza que el host se tiene que conectar a la red mediante el mismo protocolo para que le puedan enviar paquetes IP. Básicamente, esta capa emite al medio físicos los flujos de bits y

recibe los que provienen de él.

Capa de interfaz

Esta capa es la pieza clave que mantiene unida a la arquitectura. Su trabajo es permitir que los hosts envíen paquetes dentro de cualquier red y que éstos viajen a su destino de manera independiente. Como esta capa brinda un servicio no orientado a la conexión, los paquetes podrían llegar en cualquier orden y las capas superiores deberán ordenarlos.

La capa de interfaz define un paquete de formato y protocolo oficial llamado IP (Internet Protocol). El trabajo de la capa es entregar paquetes IP al destinatario. Aquí, el enrutamiento de paquetes es el aspecto principal, para evitar la congestión. Además, la capa de interfaz desencapsula los paquetes recibidos pasando a la capa superior la información.

Capa de transporte

Esta capa está diseñada para proveer comunicación extremo a extremo entre los hosts. También se encarga de coordinar múltiples aplicaciones que usan la red simultáneamente para que no se mezclen los datos, asignando identificadores a las aplicaciones.

Se utilizan dos protocolos en esta capa. El primero, TCP (Transport Control Protocol) es un protocolo confiable, orientado a la conexión que permite que un flujo de bytes que se origina en una máquina se entregue sin errores en cualquier otra máquina de la interfaz. Se divide el flujo de bytes en mensajes discretos y se los pasa a la capa de interfaz; en el receptor, se reensamblan los mensajes recibidos en el flujo de salida. TCP también maneja el control de flujo.

El segundo protocolo de esta capa, UDP (User Datagram Protocol) es un protocolo no confiable, sin conexión para aplicaciones que no necesitan la secuenciación o control de flujo de TCP. Tiene amplio uso en consultas únicas de solicitud-respuesta de tipo cliente-servidor en un solo envío, así como en aplicaciones en que la entrega puntual es más importante que la precisión (ej. transmisión de voz/video).

Capa de aplicación

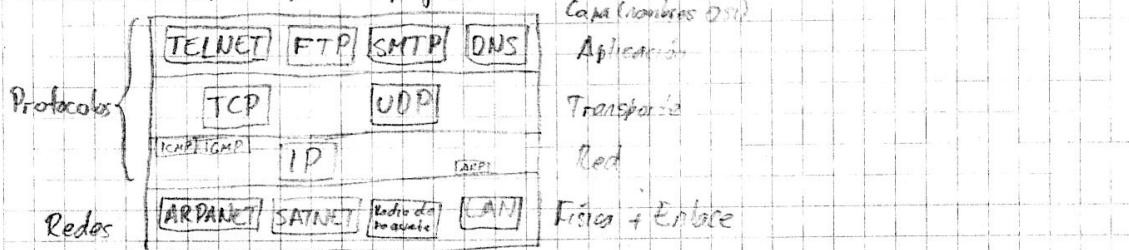
En el modelo TCP/IP no se incluyen capas de sesión y presentación, ya que la experiencia con el modelo OSI ha probado que la no inclusión es correcta, debido a su poco uso.

Entonces, arriba de la capa de transporte tenemos la capa de aplicación, que contiene todos los protocolos de nivel más alto. Los primeros incluyeron una terminal virtual (TELNET), transferencia de archivos (FTP) y correo electrónico (SMTP). TELNET permite que un usuario se registre en

Hojas 7

una máquina remota y trabaje ahí; FTP proporciona una manera eficiente de mover datos entre máquinas y SMTP es simplemente una especialización de FTP.

Con el tiempo, se han agregado otros protocolos: DNS para la resolución de nombres de host en sus direcciones de red, NNTP para transportar las noticias de USENET, SNMP para controlar y gestionar el equipo conectado a la red, HTTP para las páginas de World Wide Web, y muchos otros.



La figura anterior muestra los protocolos y redes que había inicialmente en TCP/IP.

Criticas al modelo TCP/IP

Para empezar, el modelo no distingue claramente los conceptos de servicio, interfaz y protocolo. En consecuencia, el modelo TCP/IP no es una guía para diseñar redes nuevas con tecnologías nuevas.

En segundo lugar, el modelo TCP/IP no es general del todo y no está bien ajustado para describir ninguna pila de protocolos más que de TCP/IP (ej: no podemos describir Bluetooth con TCP/IP).

Además, la capa de host a red no es en realidad una capa en el sentido normal del término, sino que es una interfaz (entre la capa de interfaz y la de enlace de datos). La distinción entre una capa y una interfaz es crucial y no se debe ser descuidada al respecto.

Por otro lado, las capas física y de enlace no son distinguibles (ni mencionadas). Un modelo adecuado debería incluir ambas capas como capas separadas, y TCP/IP no hace esto.

Por último, aunque los protocolos IP y TCP se idearon e implementaron con sumo cuidado, muchos de los demás protocolos fueron hechos con fines específicos. Posteriormente, las implementaciones de estos protocolos se distribuyeron en forma gratuita, lo que dio como resultado un uso amplio y profundo y, por lo tanto, que fueran difíciles de reemplazar (ej: TELNET).

Comparación entre los modelos

Los modelos de referencia OSI y TCP/IP tienen mucho en común. Los dos se basan en el concepto de una pila de protocolos independientes. Asimismo, la funcionalidad de las capas es muy

parecida. Por ejemplo, en ambos modelos las capas que están arriba de la capa de transporte (superiores) proporcionan un servicio de transporte independiente de extremo a extremo a los procesos que se sean comunicarse (proveedor de transporte).

A pesar de estas similitudes fundamentales, los dos modelos también tienen muchos diferencias. Para empezar, tenemos tres conceptos básicos para el modelo OSI: servicios, interfaces y protocolos. La definición de servicio indica qué hace la capa, o como funciona dicha capa (sin que la entidad superior sepa cómo lo hace). La interfaz de una capa indica los procesos que están sobre ella como accederla; específicamente cuáles son los parámetros y qué resultado se espera. Una capa es quien debe decidir qué protocolos de pares utilizar; puede usar cualesquier que desee, en tanto proporcionen los servicios ofrecidos. Originalmente, el modelo TCP/IP no distinguía estos tres conceptos, aunque se ha tratado de readaptarlo. Como consecuencia, los protocolos del modelo OSI están mejor ocultos que los del modelo TCP/IP y se pueden reemplazar fácilmente conforme cambia la tecnología.

El modelo OSI se vislumbró antes de que se inventaran los protocolos correspondientes. Esto significa que el modelo no estaba diseñado para un conjunto particular de protocolos, lo que lo hizo general. Pero los diseñadores no tenían mucha experiencia con el asunto y no tenían una idea concreta de qué funcionalidad poner en qué capa(s): la capa de enlace, que sólo trataba con redes punto a punto, y con la llegada de las redes de difusión se tuvieron que crear subcapas). Con TCP/IP ocurrió lo contrario: los protocolos llegaron primero y el modelo fue en realidad una descripción de los protocolos existentes. El problema era la poca generalidad de TCP/IP.

Una diferencia muy clara es el número de capas: el modelo OSI tiene siete y TCP/IP sólo cuatro. Los dos tienen capas de (inter)red, transporte y aplicación, pero el resto es distinto.

Otra diferencia está en el área de la comunicación orientada a la conexión contra la sin conexión. El modelo OSI soporta ambas comunicaciones en la capa de red, pero sólo la orientada a la conexión en la capa de transporte. El modelo TCP/IP sólo tiene el modo sin conexión en la capa de red, pero soporta ambos en la capa de transporte, lo que da a los usuarios la posibilidad de elegir.

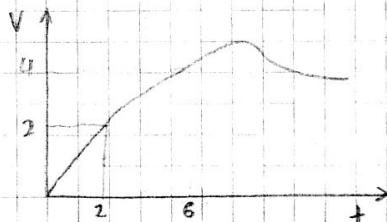
La capa física

Limitaciones físicas

Antes de estudiar las tecnologías utilizadas en la capa física, es importante conocer algunas limitaciones que impone la naturaleza y que afectan el comportamiento real de los objetos físicos.

del ideal que describe la física matemática.

La información es transmitida a través de un medio, variando alguna de sus propiedades físicas, como el voltaje o la corriente en un cable. Se le llama señal al resultado de modificar un parámetro físico del medio utilizado con la información que queremos transmitir. Representando las variaciones como una función del tiempo $f(t)$, se las puede analizar matemáticamente.



En las comunicaciones hay ciertas problemáticas fundamentales que introduce la naturaleza:

- La limitación en la frecuencia máxima a la que puede operar un medio físico, y la distorsión.
- El ruido eléctrico, electromagnético, etc.
- La velocidad de propagación de una señal a través de un medio físico es finita.

Una señal cuya forma se repite cada determinado intervalo de tiempo es llamada señal periódica. Al intervalo que medra entre repeticiones se lo denomina período (T) y a su inversa se la denomina frecuencia (f). La unidad que mide frecuencia en términos de variaciones por segundo se denomina Hertz (Hz). Se denomina amplitud a la diferencia entre el mínimo y máximo valor que puede adoptar una señal periódica.

A principios del siglo XIX, el matemático francés Jean-Baptiste Fourier probó que cualquier función periódica se puede construir con una suma (posiblemente infinita) de senos y cosenos:

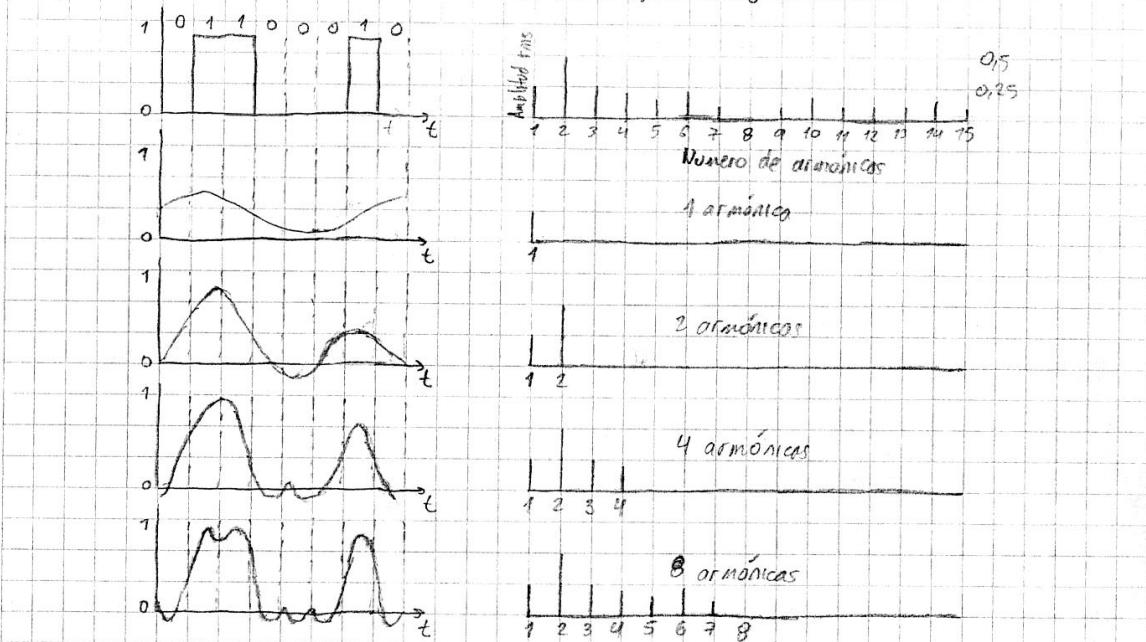
$$g(t) = \frac{c}{2} + \sum_{n=1}^{\infty} a_n \cdot \sin(2\pi n f t) + \sum_{n=1}^{\infty} b_n \cdot \cos(2\pi n f t)$$

donde $f = \frac{1}{T}$ es la frecuencia fundamental, a_n y b_n son las amplitudes de seno y coseno de los n -ésimos (términos) armónicos y c es una constante.

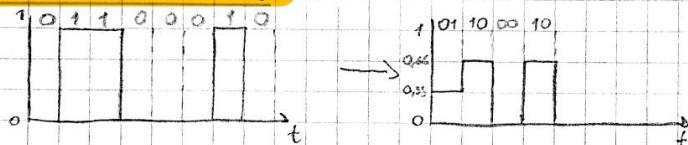
Ningún medio físico puede transmitir señales sin perder energía en el trayecto. Si todas las componentes en frecuencia de la señal se redujeron en igual proporción, dicha señal se vería reducida en amplitud, pero no distorsionada.

Sin embargo, todos los medios de transmisión reducen de distintas maneras cada una de las componentes de Fourier, y por lo tanto se produce distorsión. En general, las amplitudes se

transmitir sin disminución desde 0 hasta cierta frecuencia f_c , llamada frecuencia de corte, a partir de la cual, la atenuación aumenta considerablemente con el aumento de frecuencia. El rango de frecuencias que se transmiten con poca atenuación se conoce como ancho de banda. El ancho de banda es una propiedad física del medio de transmisión, y en algunos casos se introduce un filtro para limitar la cantidad de ancho de banda disponible (ej: teléfono).



Una técnica que permite aumentar la velocidad de transmisión en bits por segundo para en ancho de banda dado es la de codificar más de un bit por variación de la señal. Por ejemplo, si en lugar de transmitir usando dos niveles de tensión se usarán 4, se estarían transmitiendo 2 bits por cada variación, duplicando así la tasa de transmisión.



Se denomina baudio a la velocidad de variación de la señal en veces por segundo. Cuando se usan 2 niveles de señal, se da el caso particular en que la velocidad en baudios es igual a la velocidad en bps.

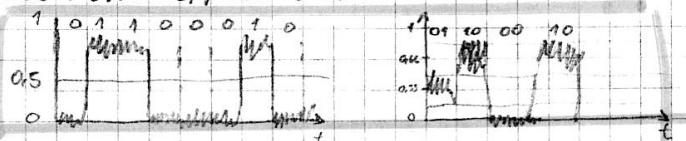
En 1924, Harry Nyquist se dio cuenta de que incluso un canal perfecto tiene una capacidad de transmisión finita. Demostros que la velocidad máxima en baudios de una señal no puede superar al doble del ancho de banda del canal por el cual se transmite.

$$C_{\text{baudios}} = 2H \rightarrow C_{\text{bps}} = 2H \cdot \log_2 V$$

$$\text{Baudio} = \text{Símbolo/s}$$

donde C_{baudios} es la máxima tasa de transmisión en baudios, C_{bps} es la máxima tasa de transmisión en bits, H es el ancho de banda del canal y V es la cantidad de niveles utilizados.

Aparentemente se podría transmitir a una gran velocidad si se utilizan muchos niveles, pero es necesario tener en cuenta el ruido. Tanto en las comunicaciones eléctricas como las radioeléctricas existe el ruido, que se traduce como perturbaciones aleatorias o pseudo-aleatorias de la señal. El ruido puede ser provocado por el hombre (chisquidos eléctricos, cierre de contactadores, interferencias de otros sistemas) o por la naturaleza (rayos cósmicos, descargas atmosféricas, explosiones solares). Existe otro tipo de ruido natural, el ruido térmico, producido por la vibración de los átomos de todo material que no se encuentre a la temperatura del cero absoluto. Este tipo de ruido es inevitable, pues es una característica intrínseca de la materia.



Es claro que si utilizamos muchos niveles, el ruido puede provocar lecturas erróneas. Entonces debemos solucionar el problema del ruido antes de ver como logramos mayor velocidad.

Para poder transmitir más rápidamente en un ancho de banda dado, es necesario aumentar la cantidad de niveles, para esto se debe aumentar el voltaje, reducir el ruido, o ambos a la vez. Se debe hacer que la potencia del ruido sea pequeña comparado con la de la señal, o sea S/N , donde se identifica a la señal y N al ruido. De otra manera queremos maximizar la relación S/N .

En comunicaciones es infrecuente hablar del cociente S/N por sí solo, se usa una unidad llamada decibel(dB), que es igual a $10 \cdot \log_{10}(S/N)$ ($S/N = 10 \rightarrow 10 \text{ dB}$, $S/N = 20 \rightarrow 20 \text{ dB} \dots$). El resultado principal de Shannon es que la tasa de datos máximas de un canal ruidoso cuyo ancho de banda es H y cuya relación señal a ruido es S/N está dada por:

$$C_{\text{máx.}} = H \cdot \log_2(1 + S/N)$$

Para aumentar la velocidad en un enlace que ya opera al límite establecido por Shannon, se debe aumentar el límite. Para esto, se debe aumentar la potencia de la señal (que dando más grande la relación S/N) obteniendo UN límite mayor al de la velocidad de operación. Luego, podemos aumentar la cantidad de niveles (bit por símbolo) de acuerdo al teorema de Nyquist.

Estos retardos de propagación pueden ser imperceptibles cuando se trata de redes locales de poca extensión, pero son notorios en casos como el de comunicaciones satelitales.

Medios existentes

Pertenecen a la capa más baja del modelo OSI y del TCP/IP, que describen las características de los distintos medios utilizados para la transmisión y define determinados parámetros de los mismos que deben tenerse en cuenta al planificar la instalación de una red de computadoras.

Estos medios se pueden clasificar en dos tipos: guidados y no guidados. Los medios poseen una tasa de transmisión, que es la cantidad de información que se transmite por unidad de tiempo. Esta está afectada por los límites físicos. La medimos en bits por segundo (bps).

Medios de transmisión guidados

Hay varios tipos, los cuales varían su ancho de banda, retraso/costo, instalación y mantenimiento.

Medios magnéticos

Es una de las formas más comunes de transportar información: almacenar datos en cintas magnéticas o medios extraíbles, transportarlos físicamente a la máquina de destino, y leerlos allí. Aunque no parece un método muy avanzado de comunicación, con frecuencia tiene mejor rentabilidad, en especial para aplicaciones en las que un ancho de banda alto o el costo por bit transportado es clave.

Por ejemplo, tenemos una cinta que almacene 70Gb y una caja que pueda contener 1000 cintas; tendríamos una capacidad de 70Tb por caja. Con un servicio de entrega que tarde 24 horas, el ancho de banda efectivo es de $560\text{ Tb} / 24\text{ hours} = 640\text{ Mbps}$; si el destino estuviera a una hora de camino, el ancho de banda aumentaría a 150 Gbps. Ninguna red de computadoras se acerca a este número.

Esto puede ser importante en bancos, donde se requiere tener centros de cómputos redundantes, y con información actualizada. Es mejor mandar un vehículo con cintas que intentar transmitir los datos usando otra tecnología de transmisión.

Naturalmente, el retraso de este medio de transmisión es bastante considerable.

UTP - Par trenzado

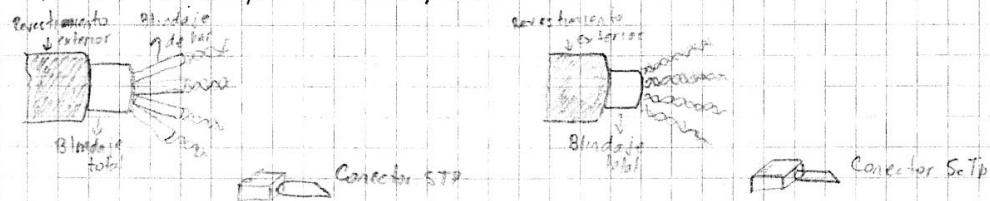
Un par trenzado está formado por dos cables de cobre con vaina de plástico reforzados uno con otro formando una trenza. Los cables se trenzan debido a que cables paralelos que forman un circuito cerrado son excelentes antenas, que captan ruido e interferencia proporcional al área encerrada entre ambos; al trenzarlos, se minimiza esta área. El UTP (Unshielded Twisted Pair) contiene cuatro pares de estos pares a su vez trenzados entre sí en una vaina plástica y

es el que se utiliza actualmente en redes Ethernet.



La primera aplicación del par trenzado fue en telefonía pero su posibilidad de uso para conexión entre terminales hizo que se usó en principio a la conexión de terminales en un mainframe, pues la posibilidad de conectar dentro de la misma vaina varios destinos permitió la misma estructura de cableado tanto para teléfono como para la conexión de terminales.

El STP (Shielded Twisted Pair) es una versión utilizada en un tiempo por IBM, que incluye un cable conductor que recubre a todos los pares aislandolos del ruido. Su costo es muy superior al de un UTP. También hay una versión más simple llamada ScTP (Screened Twisted Pair), que únicamente posee protección para todos los pares.



Los cables UTP se clasifican en:

- Categoría 3: usados en telefonía y conexiones de baja velocidad.
- Categoría 5: orientada a redes de alta velocidad.
- Categoría 6: muy reciente, para redes de muy alta velocidad.

Estas categorías se diferencian en el tipo de cobre, los materiales de aislamiento entre pares, la forma en que se trenzan y la forma en que se disponen dentro de la vaina.

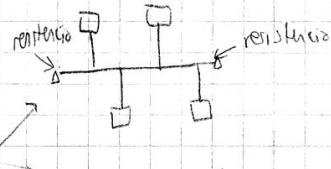
La longitud máxima de un UTP C5 es de 100m, dentro de esta distancia se aseguró que el cable mantendrá sus parámetros dentro de las especificaciones de la Categoría 5.

Cable coaxial banda base

Este cable tiene mejores prestaciones en algunos aspectos que el UTP; su forma de construcción le da mejor aislamiento a interferences y ruido eléctrico, un mayor alcance y tasa de transmisión mayor.

Se denomina de banda base haciendo referencia a un término de las comunicaciones que hace referencia a señales transmitidas sin modulación. El de uso más popular en PC es el RS-232 de 90 cm, utilizado en redes Ethernet, con un alcance de 200m.

El cable está formado, de adentro hacia afuera, por un hilo de cobre, un aislante, una malla conductora y una vaina de plástico.



Cuando se tiende una red usando coaxial, es necesario conectar cada máquina en paralelo al cable que va pasando de máquina a máquina, y clavar en ambos extremos libres del mismo una resistencia que cierra el circuito y recibe el nombre de terminador. Este era el tipo de red utilizada hasta hace varios años, debido a su bajo costo. Con el desarrollo de los hubs (que permiten usar redes con UTP), se los prefirió dadas las desventajas que presenta el coaxial.

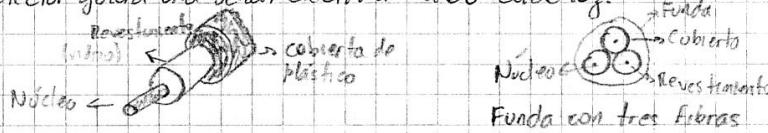
Una red de coaxial no funcionará si hay alguna interrupción en el circuito (ej: pisar el cable). Cuando esto ocurre, ninguna máquina se puede comunicar (ni siquiera las que quedaron del mismo lado del corte). Estos casos son frecuentes y es difícil localizar el origen de la falla.

Coaxial de banda ancha

A diferencia del anterior, este se diseñó para transportar señales moduladas. Su construcción permite alcanzar mayores distancias, pero sólo se lo utiliza en casos particulares.

Fibra óptica

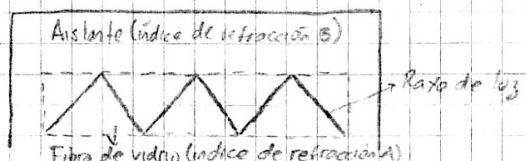
Es el medio de transmisión de mayor ancho de banda en existencia. El sistema tiene tres componentes básicos: una fuente de luz, el medio de transmisión (fibra de vidrio ultra delgado) y un detector. Un pulso de luz indica un bit en 1 y la ausencia de luz indica un bit en 0. El detector genera una señal eléctrica cuando recibe luz.



Las capacidades de la fibra óptica rondan los 50 Tbps (pruebas de laboratorio han alcanzado 100) y en aplicaciones comerciales la velocidad no supera 1 Gbps. Esta limitación se debe al equipamiento necesario para utilizar la fibra óptica, que transforma la luz en señales eléctricas y viceversa, y aún no puede alcanzar esas velocidades. Dado que la limitación no está en el

medio físico sano en la electrónica, se puede considerar a este medio como prácticamente ideal en términos de ancho de banda.

La fibra óptica hace de guía de la luz la cual avanza reflejándose con un ángulo determinado en las paredes de la misma, debido a la diferencia del índice de refracción entre el vidrio del centro y el vidrio austante.



El índice de refracción A es mucho mayor que B, esto hace que la totalidad del rayo se refleje y con esto avance por la fibra.

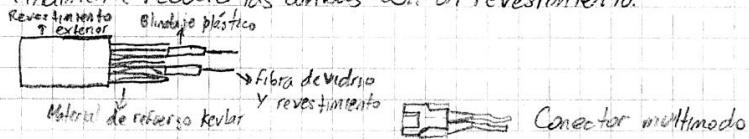
El ángulo de incidencia determina una característica de la fibra denominada modo. Existen dos tipos de fibra: los monomodo y las multimodo. La diferencia entre ambas es el diámetro de la fibra central: los monomodo son extremadamente delgados, con un diámetro comparable a la longitud de onda de la luz injectada; lo que hace que se comporte de una manera llamada "guía de ondas", que se caracteriza por transmitir la luz en línea recta (sin rebotes) y prácticamente sin atenuación. Esto hace que las fibras monomodo puedan transmitir a distancias mucho más largas sin necesidad de repetidores (varios Gbps hasta 100 km). Actualmente, la fibra monomodo es más barata que la multimodo (en metros).

Se sigue usando la fibra multimodo debido a la electrónica usada en los extremos de la fibra. La fibra monomodo requiere electrónica más compleja y usa como fuente de luz un láser, mientras que la multimodo usa electrónica menos compleja y LEDs infrarrojos para excitarse. Hay una diferencia considerable en el precio de los equipos, por lo que en las LAN se utilizan frecuentemente fibras multimodo. Los monomodo son patrimonio casi exclusivo de las grandes empresas de comunicaciones.

Como reglas útiles en el uso de fibra óptica en redes Ethernet tenemos: para Ethernet común y Fast Ethernet (10 y 100 Mbps) se puede usar multimodo para distancias hasta 2km; con monomodo algunos fabricantes aseguran hasta 70km. Para Gigabit Ethernet podemos llegar hasta 250(LED) o 450-500(LoSO)m con multimodo. Para distancias mayores es necesario el uso de monomodo.

Debido a que la fibra es un medio simplex, hace falta tener un par, uno para la trans-

misión y otra para la recepción. El par está formado por dos fibras ultra delgadas de vidrio, aisladas cada una por vidrio con un índice de refracción alto, luego recubiertas cada una con una vaina, y finalmente recubiertas ambas con un revestimiento.



Una cuestión no menor es la conectrización de la fibra, es decir, como armar los conectores que permitan empalmar una fibra con otra, o encastrarla a los equipos de los extremos, sin introducir demasiada atenuación. El empalme o conexión de fibras se puede realizar por tres métodos (es importante que las fibras en ambos extremos estén perfectamente alineadas):

- Conectores por presión: en este caso, la fibra se introduce en el conector, se alinea y después se presiona con una placa especial para ajustarla. Es un método rápido y económico, pero de poca precisión (mayor atenuación).

- Pegado Epoxy: en este caso la fibra se introduce en un conector, sellándose este con resina Epoxy, la cual una vez solidificada mantiene fija la fibra. Luego del pegado, se deben tratar perfectamente los extremos de la fibra, para mejorar su transparencia y adherencia. Con este método se logran atenuaciones tan bajas como 1dB.

- Termofusión: se utiliza cuando se deben empalmar dos fibras. Se calientan los extremos hasta que la fibra se funde y entonces se acercan hasta obtener un solo bloque de fibra. Se logra una muy baja pérdida.

Los dos primeros se usan para conectar equipos formados, el último sólo en empalmes de fibras que cubren grandes distancias y es muy costoso. Con los tres tipos de empalme pueden ocurrir reflejos en el punto de empalme, y la energía reflejada puede interferir la señal.

El emisor óptico puede ser un LED o un láser. Un LED es bastante barato, fácil de utilizar y tiene un tiempo de duración largo, pero tiene baja tasa de transmisión, cubre pocas distancias, usa fibras multimodo y es poco sensible a las temperaturas. En cambio, un láser tiene tasa de transmisión alta, cubre grandes distancias y puede utilizar ambos modos de fibra, pero posee corta vida, un costo elevado, y elevada sensibilidad a la temperatura.

Es inevitable comparar la fibra óptica con el conductor de cobre. La fibra óptica presenta un mayor ancho de banda, lo que se traduce en una mayor capacidad para transportar in-

formación, mayor tasa de transmisión; además, su baja atenuación hace que sean necesarios repetidores en distancias superiores a 10 km para tránsitos comunes, distancia que cubre perfectamente las necesidades de una MAN o LAN. Por conducir luz, no se ve afectada por el ruido eléctrico o electromagnético, y tampoco por las sustancias corrosivas del ambiente. Otra ventaja de las fibras es su diseño y poco peso (1000 cables de par trenzado de 1 km pesan 1000 kg, dos fibras tienen más capacidad y pesan 100 kg). Por último, las fibras no tienen fugas de luz y es difícil interceptarlas y conectarse a ellas, lo que le da a las fibras excelente seguridad contra espías.

Su parte negativa consiste en que es una tecnología poco familiar que requiere habilidades muy específicas. Además, las fibras pueden dañarse con facilidad si se doblan demasiado. Por último, las interfaces de fibra son más costosas que las eléctricas.

Medios de transmisión no guidados - Transmisión inalámbrica

Cuando el tipo de terreno, las dificultades de instalación o la movilidad de los puntos de conexión no permiten el tendido de cables, es necesario utilizar otro tipo de medio. Esta otra forma es la transmisión inalámbrica, que utiliza las ondas electromagnéticas para el transporte de la información.

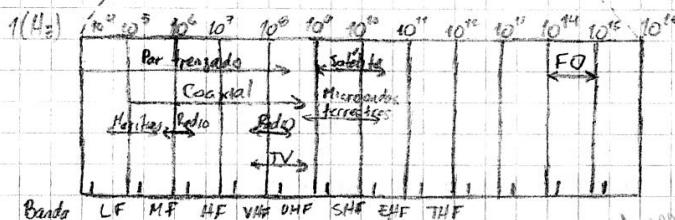
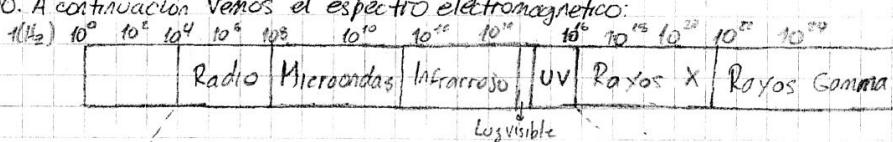
Cuando los electrones se mueven crean ondas electromagnéticas que pueden propagarse por el espacio libre (incluso en el vacío). El Físico alemán ^{Heinrich Hertz} las observó en 1887. La cantidad de oscilaciones por segundo de una onda electromagnética es su frecuencia f , que se mide en Hz. La distancia entre dos máximos (máximos) se llama longitud de onda y se designa con λ .

Al conectar una antena del tamaño apropiado a un circuito eléctrico, las ondas electromagnéticas pueden ser difundidas de manera eficiente y ser captadas por un receptor a cierta distancia. Toda la comunicación inalámbrica se basa en este concepto.

En el vacío, todas las ondas electromagnéticas viajan a la misma velocidad, sin importar su frecuencia. Esta velocidad es la de la luz, $c = 3 \times 10^8$ m/s. En el cobre o la fibra óptica, la velocidad baja a casi 2/3 de este valor y se vuelve ligeramente dependiente de la frecuencia.

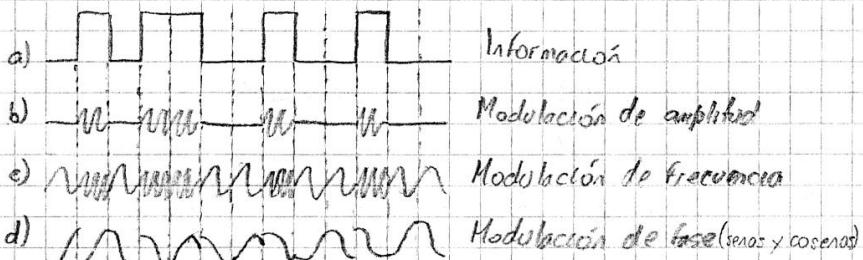
La relación fundamental entre f , λ y c (en el vacío) es $\lambda f = c$. Puesto que c es constante, dado λ podemos encontrar f y viceversa. Como regla general, si λ se expresa en metros y f en MHz,

$\lambda f \approx 300$. A continuación vemos el espectro electromagnético:



La luz ultravioleta, los rayos X y los rayos gamma serían buenas transmisoras, pero son difíciles de producir y modular, no se propagan bien en edificios y son peligrosos para los seres vivos.

Para poder transmitir información a través de las ondas electromagnéticas, éstas deben ser moduladas usando su amplitud, su frecuencia o su fase.



En general las transmisiones ocupan una banda estrecha de frecuencias, o sea que se utiliza una frecuencia determinada como portadora, y la modulación utilizada varía la frecuencia de la portadora en valores relativos muy pequeños. Este método se prefiere porque se obtiene una mejor recepción (mas W por Hz).

Sin embargo, también se utiliza la dispersión de la transmisión en una banda ancha de frecuencia. Esta técnica se denomina espectro disperso y se utilizó inicialmente en comunicaciones militares, pero actualmente se utiliza ampliamente en aplicaciones comerciales (Ethernet sobre fibra óptica).

Ondas de radio

Las ondas de radio son fáciles de generar, pueden viajar largas distancias y beneficiar edificios sin problemas. También son omnidireccionales, lo que significa que viajan en todas las direcciones a partir de la fuente. Tienen el problema de que se reduce la potencia drásticamente a grandes distancias, especialmente en bajas frecuencias.

La interferencia entre ondas es un problema, por eso todos los gobiernos reglamentan estrictamente el uso de radio transmisores. El uso de estas bandas para la transmisión de

datos no es muy conveniente, ya que poseen un ancho de banda limitado.

Microondas

Por encima de los 100 MHz las ondas viajan en linea recta y se pueden enfocar en un haz estrecho. Al concentrar toda la energía en un haz pequeño con una antena parabólica se produce una señal mucho más alta en relación con el ruido, pero las antenas transmisoras y receptoras deben estar muy bien alineadas entre sí. Además esta direccionalidad permite tener varios transmisores en fila conectándose con varios receptores en fila, sin interferencia.

A diferencia de las ondas de radio, los microondas no atraviesan objetos. Además, aún cuando el haz pueda estar bien enfocado en el transmisor, hay cierta divergencia en el espacio. Algunas ondas pueden refractarse en las capas atmosféricas más bajas y tardar un poco más en llegar que las ondas directas; luego pueden llegar fuera de fase y cancelar la señal, produciendo un efecto conocido como desvanecimiento por múltiples trayectorias. En frecuencias altas, el clima (particularmente la lluvia) se comporta como una barrera que absorbe la señal.

Este tipo de ondas se utiliza mucho en los sistemas telefónicos de larga distancia, los teléfonos celulares, la distribución de la televisión, etc. También se usa en comunicaciones de datos punto a punto, usando tecnologías tradicionales o la técnica de espectro disperso.

Infrarrojos

Se utilizan mucho en las comunicaciones de corto alcance. Son relativamente direccionales, baratos y fáciles de producir, pero no atraviesan objetos sólidos (controles remotos).

Por otro lado, el hecho de que no atraviesen paredes puede ser una ventaja, ya que varios sistemas similares que estén en diferentes casas no interferirán entre sí. Esto hace que su seguridad contra el espionaje sea superior a la de los sistemas de radio. Además, no es necesario obtener una licencia gubernamental para operar un sistema infrarrojo.

Luz

La forma más común de utilizarla es con rayos láser y foto-detectores. Por ejemplo, se pueden conectar dos LAN en edificios separados por medio de lentes montados en sus azoteas. La señalización óptica entre los láseres es inherentemente unidireccional, de modo que cada edificio necesita

síntesis su propio láser y su propio foto-detectador. Este dispositivo ofrece un ancho de banda muy alto y un costo muy bajo, además de ser relativamente fácil de instalar.

Sin embargo, la ventaja del láser, con haz muy estrecho, es también una debilidad, ya que se debe ser muy preciso al apuntar. En general, se añaden lentes al sistema para desenfocar ligeramente el rayo. Otro problema es que no puede penetrar objetos y es muy influido por el clima.

Comunicación satelital

La idea detrás de la comunicación satelital es la siguiente: se coloca un satélite artificial en órbita. Desde un punto de la Tierra se transmite una señal electromagnética hacia el satélite; luego pueden pasar dos cosas: la señal rebota y es recibida en otro punto de la superficie, o la señal es recibida por el satélite, amplificada y retransmitida en otra parte del espacio hacia otro punto.

Satélites geosintonicos

A una altitud aproximada de 36.000 km sobre el ecuador, el período del satélite es de un día sideral, de modo que gira a la misma velocidad que la Tierra bajo él. Es un extremo deseable tener el satélite fijo en el cielo, ya que sino se necesitaría una costosa antena girable.

Con la tecnología actual, es poco recomendable utilizar satélites geostacionarios espaciados a menos de dos grados en el plano ecatorial para evitar interferencia, lo que nos deja en máximo de 180 satélites. Para evitar el caos, la ITU asigna la posición orbital a los países y también significa bandas de frecuencia específicas a los usuarios de satélites. Para permitir que el tráfico fluya en ambos sentidos al mismo tiempo, se requiere un canal para cada sentido.

Banda	Frecuencias	Enlace desc.	Enlace asc.	Ancho	Problemas
L	0.9/1.02	1,5	1,6	15 MHz	Bajo ancho de banda, saturado
S	0.4/1.02	1,9	2,2	70 MHz	Bajo ancho de banda, saturado
C	0.4/6.6/10.2	4	6	500 MHz	Interferencia terrestre
Ku	11/14/10.2	11	14	500 MHz	Lluvia
Ka	20/30	20	30	3500 MHz	Lluvia, costo del equipo

La banda C fue la primera que se destinó al tráfico comercial por satélite y actualmente está superpoblada. Los bandas L y S fueron incorporadas mediante un acuerdo internacional, pero son estrechas y saturadas. La siguiente banda disponible, Ku, aún no está saturada, pero la lluvia presenta un gran problema ya que el agua absorbe estas microondas. Este problema se soluciona

con la instalación de varias estaciones terrestres, pero esto aumenta el costo. También se asignó la banda Ka, pero el equipo necesario para utilizarla todavía es caro. Además de estas bandas existen muchas otras gubernamentales y militares.

Un satélite moderno tiene alrededor de 40 equipos de comunicaciones (llamados transponders), cada uno con un ancho de banda de 80 MHz.

Los satélites de comunicaciones tienen propiedades diferentes de los enlaces terrestres punto a punto. Para empezar, el retraso de una estación terrestre a otra, pasando por un satélite puede ser de entre 250 y 300 ms, comparado con un enlace terrestre por microondas que tiene un retraso de casi 3 ns/km, o enlace de cable coaxial o fibra óptica que tienen retraso de 5 ns/km.

Otra propiedad importante de los satélites es que son esencialmente medios de difusión, por lo que no cuesta más mandar un mensaje a miles de estaciones dentro del alcance de un transmisor que mandarlo a una sola. Esta propiedad es muy útil en algunas aplicaciones, pero en materia de seguridad y privacidad es malo ya que cualquiera puede escuchar todo. La encriptación es esencial cuando se requiere seguridad.

Los satélites también tienen la propiedad de que el costo de transmitir un mensaje es independiente de la distancia que se recorra. También cuentan con excelentes tasas de error y se pueden instalar en forma casi instantánea.

Otros satélites

Al estar limitado el número de satélites geoestacionarios, también se utilizan satélites a otras altitudes. Los satélites MEO tardan 6 horas en dar vuelta a la Tierra (ej: GPS). Los LEO están mucho más cerca de la superficie y se utilizan para proyectos como Iridium (que busca cobertura global para telefonía celular), Globalstar o Teledesic.

Satélites y fibra

En los '70 se pensaba que el futuro de las comunicaciones era por satélite. Pero la introducción de la competencia entre proveedores de comunicaciones cambió todo, ya que las compañías telefónicas comenzaron a reemplazar por fibra sus redes de largo alcance e introdujeron servicios de alto ancho de banda. Entonces, parecía que las conexiones terrestres de fibra ganarían el ganador a largo plazo.

Pero los satélites tienen algunos nichos de mercado en los que la fibra no tiene presencia:

- Aunque en principio una sola fibra tiene más ancho de banda potencial que todos los satélites que se han lanzado, este ancho de banda no está disponible para todos los usuarios. Los fibra se utilizan en el sistema telefónico actualmente para manejar muchas llamadas de larga distancia al mismo tiempo, no para ofrecer un ancho de banda alto a los usuarios individuales. Con los satélites, es factible instalar una antena en el techo y evadir el sistema telefónico para conseguir un ancho de banda alto.
- En la comunicación móvil los enlaces por fibra no son adecuados, aunque es posible que una combinación de radio celular y fibra funcione. Sin embargo, es mejor usar satélites.
- Cuando la difusión es esencial, un mensaje enviado por satélite puede ser recibido por miles de estaciones terrestres, y esto es más económico que simular la difusión en tierra.
- En lugares con terreno escabroso o infraestructura terrestre poco desarrollada, un satélite puede ser mucho más apropiado y económico que colocar fibras o conexiones punto a punto.
- Otros casos es cuando obtener el derecho de paso para tender la fibra en un terreno es difícil o excesivamente costoso.
- Cuando un despliegue rápido es primordial (sistema de comunicaciones militares en tiempos de guerra), los satélites ganan con facilidad.

En síntesis, parece que la corriente principal en la comunicación del futuro será la fibra óptica terrestre combinada con radio celular, pero para algunos usos especializados los satélites son mejores. Sin embargo, hay un imponente que se aplica en todos los casos: el aspecto económico. Aunque la fibra ofrece mayor ancho de banda, es posible una competencia agresiva en precios.

El sistema telefónico

Al tener dos computadoras ubicadas una cerca de la otra, con frecuencia lo más fácil es conectarlas mediante un cable (así funcionan las LAN). Sin embargo, cuando las distancias son muy grandes, o hay muchas computadoras, o los cables tendrían que pasar por una vía pública o una zona restrictiva, los costos de tender cables privados por lo general son prohibitivos. Además, en casi todos los países también es ilegal instalar líneas de transmisión privadas a través (o debajo) de una propiedad pública. Por lo tanto, los diseñadores de redes a veces deben apoyarse en las instalaciones de telecomunicaciones existentes.

Por lo general, estas instalaciones fueron diseñadas hace muchos años, con un propósito totalmente

Viejo: pero básicamente que la fibra se multiplexa mucho y al usuario final le llega menos ancho de banda efectivo. En cambio una antena es toda para vos

distantes transmitir la voz humana en una forma más o menos reconocible. Su aplicabilidad en las comunicaciones de computadora a computadora es muy limitada, pero esta situación está cambiando con la introducción de la fibra óptica y la tecnología digital.

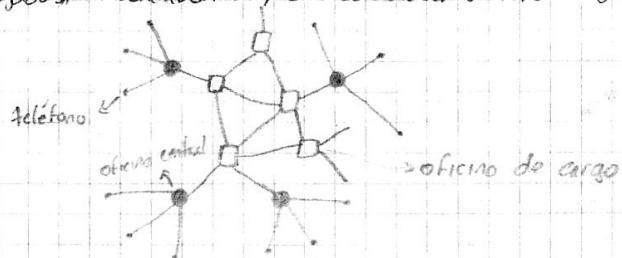
Estructura del sistema telefónico

La estructura se organiza en una forma jerárquica altamente redondeante, de múltiples niveles. Daremos una descripción simplificada para que se comprenda la idea esencial.

Cada teléfono está conectado por un par trenzado de cobre a la oficina central más cercana de la compañía telefónica (también llamada oficina central local). Por lo general la distancia es de 1 a 10 km, y en las ciudades es menor que en las áreas rurales. Estas conexiones se conocen como lazo local o vínculo local. La concatenación del código de área y los tres primeros dígitos del número telefónico es específica de manera única una oficina central, y por ello la estructura de tarifas se basa en esta información.

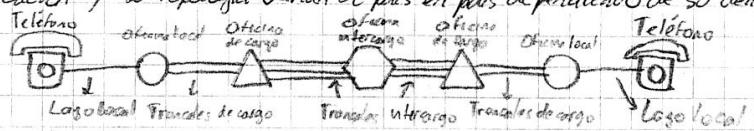
Si un suscriptor conectado a una oficina central determinada llama a otro suscriptor conectado a la misma oficina central, el mecanismo de conmutación dentro de la oficina establece una conexión eléctrica directa entre los dos lazos locales. Esta conexión permanecerá intacta mientras dure la llamada.

Si el teléfono al que se llamará está conectado a otra oficina central, se tiene que usar un procedimiento diferente. Cada oficina central tiene varias líneas subterrenas a uno o más centros de conmutación cercanos, llamados oficinas de cargo interurbanas. Estas líneas se llaman troncales de conexión con cargo. Si sucede que tanto la oficina central de quien llama como la de quien es llamado tienen un troncal de conexión a la misma oficina de cargo (algo muy probable si no están alejadas), la conexión se puede establecer dentro de la oficina de cargo.



Si el que llama y el que es llamado NO tienen una oficina de cargo en común, la trayectoria se deberá establecer en un nivel más alto de la jerarquía. Hay oficinas primarias, secundarias y re-

cionales que forman una red que conecta a las oficinas de cargo. Todas estas se comunican entre sí mediante troncales intercargos de gran ancho de banda. La cantidad de fibras diferentes de centros de comunicación y su topología varían de país en país dependiendo de su densidad telefónica.



Para telecomunicaciones se usan diversos medios de transmisión. En general, para los lazos locales se utilizan pares trenzados, con señalización analógica. Entre las oficinas de conmutación se usan ampliamente cables coaxiales, microondas y, más que nada, fibra óptica, con señalización digital.

Antiguamente, la señalización en todo el sistema telefónico era analógica, es decir que la señal de la voz real era transmitida como un voltaje eléctrico del origen al destino. Ahora, con la electrónica digital y los computadores, se usa señalización digital en los troncales entre oficinas de conmutación, mientras que el lazo local queda como el único elemento analógico del sistema.

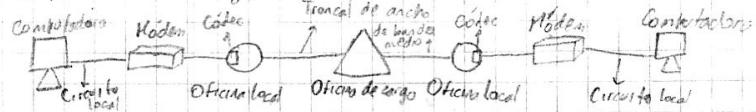
La señalización digital tiene muchas ventajas sobre la analógica:

- Aunque la atenuación y la distorsión son más severas cuando se envían señales de dos niveles, es fácil calcular qué tan lejos se puede propagar una onda y todavía ser reconocible. Se pueden utilizar regeneradores digitales en esos puntos de la línea para restablecer la señal a su valor original. Una señal digital puede pasar a través de una cantidad arbitraria de regeneradores sin pérdida de señal y viajar así grandes distancias sin pérdida de información, a diferencia de las señales analógicas que sufren pérdidas al amplificarse.
- Se puede utilizar para transmitir voz, datos, música e imágenes para aprovechar de forma más eficiente los circuitos y el equipo.
- Es más económica pues no es necesario reproducir exactamente una forma de onda analógica desviéndola a través tal vez cientos de amplificadores en una llamada transcontinental, es suficiente con distinguir correctamente cada uno de un tramo.
- El mantenimiento es más simple, ya que en la transmisión se recibe o no, con lo que se facilita el rastreo de problemas.

El lazo local

Debido a que los lazos locales todavía son analógicos, es necesario que los datos digitales generados por la computadora sean transformados a una forma analógica por un módem para

poder ser transmitidos por la línea de disco, luego convertirse a la forma digital para transmitirse por los troncales de largo alcance, después reconvertirse a analógicos en el lazo local del extremo receptor y, por último, a digitales con otro módem para almacenarse en la computadora de destino.



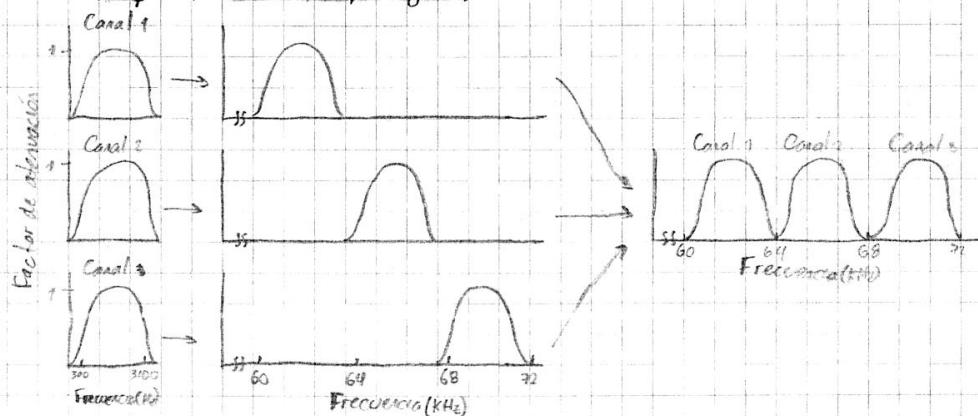
En el caso de líneas rentadas es posible trabajar en forma digital de principio a fin, pero estas líneas son caras y sólo sirven para construir redes privadas entre compañías.

Para realizar la conversión de digital a analógico, los módems utilizan los métodos de modulación mencionados en los medios de transmisión inalámbricos.

Troncales y multiplexación

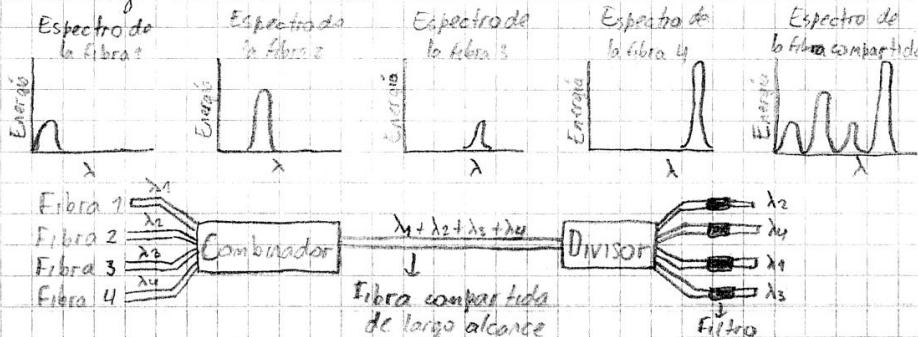
Desde el punto de vista económico, instalar y mantener un troncal de bajo ancho de banda cuesta prácticamente lo mismo que uno de alto ancho de banda, esto es porque el gasto principal es el dado por la excavación de zanjas y no el cable o la fibra óptica. En consecuencia, las compañías telefónicas han desarrollado esquemas elaborados para multiplexar muchas conversaciones en un solo troncal físico. Estos se pueden dividir en dos categorías: FDM y TDM.

Frequency division multiplexing (FDM)



Primero se utilizan filtros para limitar el ancho de banda utilizable a cerca de 3kHz por canal de grado de voz. Cuando se multiplexan muchos canales juntos, se asigna 4kHz a cada canal para mantenerlos bien separados. Se eleva la frecuencia de los canales de voz, cada uno en una cantidad diferente después de lo cual ya se pueden combinar, porque ahora no hay dos canales que ocupan la misma porción del espectro.

Para los canales de fibra óptica se utiliza una variante de la FDM llamada multiplexación por división de longitud de onda (WDM).

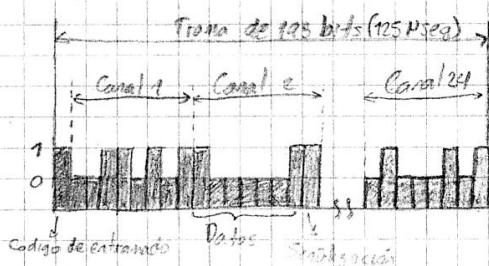


Los distintos cables se juntan en un combinatorio óptico, cada uno con su energía presente a diferentes longitudes de onda. Los cables se combinan en una sola fibra compartida para transmisión a un destino distante, donde el haz se divide en tantas fibras como hayan entrado. Cada fibra saliente contiene un núcleo corto especialmente construido que filtra todas las longitudes de onda, salvo una.

En realidad, esto no es nada nuevo; es FDM a frecuencias muy altas. La única diferencia con la FDM eléctrica es que en sistema óptico que usa una rejilla de difracción es totalmente pasivo, y por ello, altamente confiable.

Time division multiplexing (TDM)

A diferencia de la FDM, la TDM se puede controlar completamente con electrónica digital, de modo que se ha extendido de forma amplia en años recientes. Por desgracia, sólo se puede utilizar con datos digitales. Lo que se hace es digitalizar con un códec la señal analógica que viene por el cable local (técnica PCM), y luego se combinan en una única señal digital de salida. Por ejemplo, el formato DS1 contiene 24 canales de voz que se multiplexan juntos; cada uno de los canales inserta 8 bits por turno en la corriente de salida.



Oficinas de conmutación

Hasta ahora hemos visto lo que se llama la planta externa, es decir, los elementos que van por fuera de las oficinas. Se denominan planta interna a todos los dispositivos utilizados dentro de las oficinas, o sea los conmutadores.

Para realizar la conmutación se pueden usar las dos técnicas anteriormente vistas: conmutación de circuitos y conmutación de paquetes.

Módems y velocidad máxima

Las líneas telefónicas analógicas tienen una frecuencia de corte introducida artificialmente que ronda los 3 kHz. Si no se introduce ese límite, es posible transmitir sobre el par teléfono a velocidades bastante altas, siempre que las distancias sean cortas. Con esta frecuencia de corte, según Shannon no sería posible transmitir más que 2400 bps; las líneas actuales no pueden transmitir a más de 33600 bps.

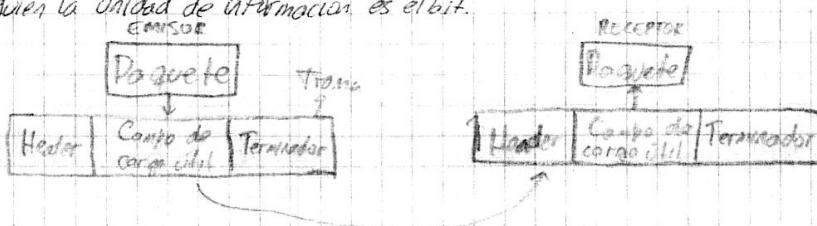
Para poder tener módems de 56K es necesario tener líneas no "tan" analógicas. La forma de aumentar la velocidad es bajar el ruido y mejorar la calidad de la señal. Esto se logra suprimiendo el bucle de abonado analógico de uno de los extremos, reemplazándolo con uno digital.

Últimamente se ofrece a los usuarios de Internet el servicio de línea de abonado digital (DSL), que consiste en multiplexar en un par de cables un canal digital de telefonía (64 Kbps) y uno de datos de 512. En nuestro país, el canal de datos que va al domicilio del cliente es más veloz que el que va hacia la central, por lo tanto el servicio se denomina asimétrico, de ahí el nombre de ADSL.

La capa de enlace

La capa de enlace controla la comunicación directa por el medio físico entre dos máquinas adyacentes, implementando los algoritmos necesarios para que la información se transfiera en el orden que se generó, libre de errores, a una velocidad compatible con ambas máquinas y controlando la forma en que se accede al medio físico. También define el formato de la trama y el direccionamiento de hardware.

En general, se habla de trama cuando la unidad de información transmitida es todo bit y paquete cuando la unidad de información es un conjunto de bits. La capa de enlace generalmente toma paquetes de la capa de red y los convierte en tramas para enviarlos a la capa física, para quien la unidad de información es el bit.



2 Capa de enlace

Capa de enlace

Subcapa de control de enlace

Subcapa de acceso al medio

IEEE 802.3 - Ethernet

IEEE 802.11 - Wifi

Para realizar la conmutación se pueden usar las dos técnicas anteriormente vistas: conmutación de circuitos y conmutación de paquetes.

Módems y velocidad máxima

Las líneas telefónicas analógicas tienen una frecuencia de corte introducida artificialmente que ronda los 3 kHz. Si no se introduce ese límite, es posible transmitir sobre el par telefónico a velocidades bastante altas, siempre que las distancias sean cortas. Con esta frecuencia de corte, según Shannon no sería posible transmitir más que 2400 bps; las líneas actuales no pueden transmitir a más de 33600 bps.

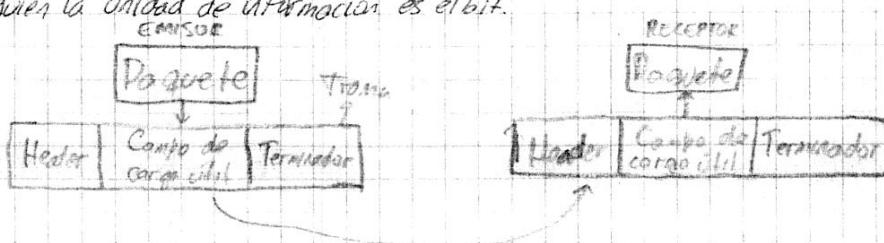
Para poder tener módems de 56K es necesario tener líneas no "tan" analógicas. La forma de aumentar la velocidad es bajar el ruido y mejorar la calidad de la señal. Esto se logra suprimiendo el bucle de abono analógico de uno de los extremos, reemplazándolo con uno digital.

Últimamente se ofrece a los usuarios de Internet el servicio de línea de elevado digital (DSL), que consiste en multiplexar en un par de cable un canal digital de telefonía (800 Kbps) y uno de datos de 512. En nuestro país, el canal de datos que va al domicilio del cliente es más veloz que el que va hacia la central, por lo tanto el servicio se denomina asimétrico, de ahí el nombre de ADSL.

La capa de enlace

La capa de enlace controla la comunicación directa por el medio físico entre dos máquinas adyacentes, implementando los algoritmos necesarios para que la información se transfiera en el orden que se generó, libre de errores, a una velocidad compatible con ambas máquinas y controlando la forma en que se accede al medio físico. También define el formato de la trama y el direccionamiento de hardware.

En general, se habla de trama cuando la unidad de información transmitida está orientada al bit y paquete cuando la unidad de información es un conjunto de bits. La capa de enlace generalmente toma paquetes de la capa de red y los convierte en tramas para enviarlos a la capa física, para quien la unidad de información es el bit.



Cuando se diseñó originalmente el modelo OSI, no se tuvieron en cuenta las redes de difusión y la gran cantidad de funciones específicas que acarrea controlar el acceso al medio en este caso, por lo que posteriormente se realizó una división en dos subcapas para separar tareas que se pueden agrupar en diferentes categorías: LLC y MAC.

El servicio principal otorgado a la capa de red es la de transferencia de datos entre dichas capas desde la máquina de origen a la de destino. Hay 3 formas razonables de ofrecer este servicio:

- Servicio sin acuse ni conexión: consiste en hacer que la máquina de origen envíe tramas independientes a la máquina de destino sin pedir que ésta los reconozca o acuse su recepción. No se establece conexión de antemano ni se libera después. Este servicio es apropiado cuando la tasa de errores es baja, por lo que se deja la recuperación a los capas superiores. También es apropiado para el tráfico en tiempo real, por ej. de voz, en el que la llegada retardada de datos es peor que los errores. La mayoría de las LAN utilizan esta clase de servicio.

- Servicio con acuse sin conexión: es una optimización conveniente cuando se utilizan medios de transmisión inestables, como el inalámbrico. En este caso cada trama es reconocida individualmente y se envía un acuse de recepción al origen cuando llega al destino. Cabe aclarar que en los medios confiables no es conveniente este método, ya que solo hay errores ocasionalmente, por lo que se desperdiciaría ancho de banda en enviar acuses por cada trama. En estos casos, los (pocos) errores pueden ser corregidos sin problemas por las capas superiores.

- Servicio con acuse y orientado a la conexión: con este servicio, las máquinas de origen y destino establecen una conexión antes de transferir datos. Cada trama enviada a través de la conexión está numerada, y se garantiza que llegarán en orden y sin errores. Este tipo de servicio se usa frecuentemente en líneas punto a punto, ya sea por medios inalámbricos o líneas telefónicas.

Subcapa de control de enlace (LLC)

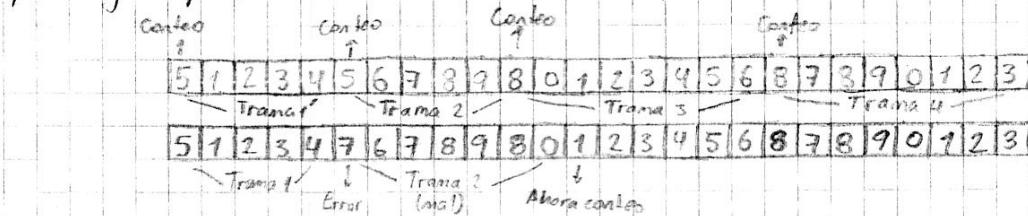
Esta subcapa es la parte superior de la capa de enlace, y controla servicios como reordenamiento de tramas para devolverlos datos como fueron entregados, control de flujo, y control de errores.

Enmarcado de la trama

La capa de la enlace debe enviar y recibir datos desde la capa de enlace de la máquina adyacente. Para hacer eso, construye tramas y las envía a la capa física. Es claro que es necesario que el receptor pueda detectar con precisión donde empieza y termina cada trama, es decir, se deben delimitar

tar los tramos. Esta delimitación debe ser independiente de la temporización, por lo tanto deben definirse reglas para enmarcar los datos sin necesidad de depender del tiempo. Tenemos varios métodos para hacer esto, veremos los cuatro más comunes.

El primer método es la cuenta de caracteres. Se inserta un campo en la trama que tiene la longitud en caracteres de la misma. El problema de este método, que hace que se lo utilice pocas veces, es la pérdida del sincronismo cuando existe un error en el campo que contiene la longitud, lo que obliga a pedidos de retransmisión frecuentes y incluso esto no lo soluciona.



El segundo método es el de caracteres de inicio y parada con inserción de caracteres. Cada trama comienza y termina con una secuencia de caracteres determinadas (caracteres de enlace), por lo tanto se descarta el problema del error en la longitud, pues la trama está perfectamente enmarcada y no hay necesidad de conocer de antemano la longitud; sólo se tiene que buscar el carácter de fin de trama. El problema radica en que los caracteres de enlace pueden confundirse con los datos, en particular si se transmiten datos en código binario puro. Esto se soluciona agregando un carácter (llamado de escape) antes de cada carácter de enlace, y si se quiere marcar el carácter de escape entre los datos, simplemente se lo duplica. La desventaja de este método es su dependencia con el conjunto de caracteres usado por cada máquina en particular.

El tercer método es el de banderas de inicio y final con inserción de bits, que soluciona el problema del anterior al trabajar al nivel de bits, independizándose de la longitud de los caracteres. Se hace algo similar al caso anterior, pero a nivel de bits. Por ejemplo, se convierte que la secuencia 0-111110 se interpreta como delimitador. Si se quiere marcar esta secuencia como dato, se aplica lo siguiente: si dentro de los datos tienen más de 5 bits seguidos en 1, se inserta en 0 luego de cada quinto bit en 1. El receptor, cada vez que vea cinco 1 seguidos de un 0, remueve este último.

El cuarto método se denomina violación de código de capa física y requiere que a nivel físico se transmita una señal cuya codificación sea distinta de la utilizada para representar el

O es el 1. Por ejemplo algunos LAN codifican 1 con un par alto-bajo y 0 con un par bajo-alto. El ancho que se logra introduciendo las combinaciones alto-alto y bajo-bajo es violento la codificación de la capa física.

Muchos protocolos utilizan la combinación de una cuarta de caracteres con uno de los otros métodos como una seguridad extra.

Control de errores

El control de errores cubre tanto la detección como la corrección de los mismos. Algunos protocolos proveen mecanismos para que además de conocer que se ha producido un error (detección), se pueda recuperar cuál fue la información que realmente se quiso transmitir (corrección). Los protocolos que sólo detectan requieren retransmisión de la información.

La decisión entre detección de errores y pedido de retransmisión o detección y corrección sin retransmisión responde a las condiciones en que se establece la comunicación. Si se desea poder corregir un error, se tiene que transmitir información redundante, que debe ser mayor que aquella que sólo me permite saber que hubo un error y nada más.

Si el canal utilizado es un canal lento, por ejemplo una línea telefónica, se justifica un algoritmo de corrección de errores, como Hamming. Este algoritmo establece datos redundantes a transmitir junto con los datos originales, de forma que si hubo errores en los datos, éstos se pueden reconstruir a partir de los datos redundantes.

Si el canal disponible presenta una velocidad media o alta, se recurre a la detección y pedido de retransmisión como solución a los errores de transmisión, para mayor eficiencia. Un algoritmo muy difundido y utilizado, en particular por la norma IEEE802, es el CRC (Código de Redundancia Cíclica). Los CRC pueden detectar errores en tramas muy grandes, ocupando tan solo 2 octetos.

Control de flujo

Para evitar que máquinas rápidas o con gran cantidad de información para transmitir sobre-carguen la capacidad de recibir datos de la máquina destino, se debe introducir un control de flujo. Este control generalmente requiere de un mecanismo de retroalimentación, para que el transmisor pueda enterarse si el receptor es capaz de mantener el ritmo o no. Otro mecanismo es limitar la tasa a la que el emisor puede transmitir los datos.

Se conocen varias esquemas de control de flujo, pero la mayoría se basa en el mismo prin-

cípico. Se establecen reglas que impiden que el transmisor envíe datos hasta que el receptor lo haya autorizado, implícita o explícitamente. Por ejemplo, el receptor puede pedir la transmisión de tramas, pero una vez transmitidas, no se puede volver a transmitir hasta que el receptor lo ordene.

Algunos protocolos de enlace de datos

- **HDLC** (High-level Data Link Control) es un protocolo confiable y no orientado a la retransmisión. Tiene la posibilidad de detectar errores, pedir retransmisiones y varias funciones más. Este protocolo se inventó en la época que las líneas punto a punto eran poco confiables y muy frágiles a comunicar errores. Además, el hardware era bastante caro y muchas veces se utilizaban equipos con poca capacidad de procesamiento y poca inteligencia. En este contexto, valió la pena que la red proporcionase confiabilidad y le entregue el problema resuelto a los equipos de los extremos.

- **PPP** (Point to Point Protocol): se encapsulan los datos de la capa de red para transmitirlos por una línea serie, además se proveen funciones de detección de errores y compresión de paquetes, otorgando un servicio con acceso orientado a la conexión. Es un protocolo muy difundido, por ejemplo, es el que se utiliza cuando nos conectamos por móvil a Internet.

- **Frame Relay**: en los últimos tiempos las comunicaciones digitales punto a punto se fueron haciendo muy confiables, y el hardware más barato y con mayor potencia de procesamiento. Los requerimientos pusieron a ser transmisión de datos lo más rápido posible y a bajo costo. Se creó el protocolo Frame Relay que no tiene ~~esa~~ ningún tipo de control de errores. Este servicio es prestado por las empresas de telecomunicaciones y consiste en crear circuitos virtuales entre las distintas redes del cliente. Desde una ubicación, parten varios circuitos virtuales (por el mismo cable), ya que cada trama lleva en campo con el número del circuito virtual al que va destinado. Hay muy poco aporte de este campo de dirección, lo que hace que el protocolo sea muy liviano. En los pocos casos en que se produzcan errores su tratamiento es responsabilidad de las capas superiores.

Representa un ahorro de costo ya que en lugar de tener que disponer de una línea dedicada entre cada ubicación de cliente, sólo se requiere que dichas ubicaciones estén conectadas a la red del proveedor. Además, el costo de enlace de Frame Relay es independiente de la distancia.

Al trabajar con Frame, se puede transmitir a velocidad máxima en ráfagas cortas, pero el frame

dio a largo plazo se obtiene debido de cierto valor llamado CIR (Committed Information Rate). De todos maneras, muchos nodos transmiten en ráfagas, y esto permite utilizar este protocolo para bajar el costo y aun así brindar un buen rendimiento.

Sobrepila de acceso al medio (MAC)

Vemos que la forma de transmitir datos entre máquinas se puede dividir en dos tipos: redes punto a punto o redes de difusión. En las últimas, el asunto clave es la manera de determinar quién puede usar el canal cuando hay competencia por él. La MAC se introdujo al definirse la norma IEEE802 y tiene especial importancia en las LANs. Sus protocolos definen las reglas para arbitrar el uso del vínculo físico.

Direcciones de hardware de red

Cada tecnología usada para el hardware de red debe definir un mecanismo de direccionamiento que se utiliza para especificar el destino de cada paquete. A cada computadora de la red se le asigna una única dirección y cada paquete enviado incluye el destino del mismo.

Cada tecnología especifica como las computadoras son asignadas a una dirección. El hardware especifica, por ejemplo, el número de bits en la dirección así como la localización del campo de dirección de destino en un paquete. A esta dirección se la llama dirección MAC, ya que es esta subcapa la que fija o ignora cada paquete dejándole de su dirección. Entre estas direcciones existe uno especial llamada de difusión o broadcast, la cual es procesada no importa cuál sea la dirección de la máquina. También se presenta la posibilidad de que mediante direccionamiento especial se abra que un determinado rango de direcciones.

Topologías de LAN

Ya hemos visto disposiciones físicas de una red que se refieren a la forma de tender el cableado entre máquinas (anillo, bus, malla, estrella); esto lo conocemos como la topología física de una red.

También tenemos estructuras lógicas de redes, que se refieren a los protocolos de arbitraje para clasificarse al medio; en las redes de difusión actuales encontramos las topologías en bus y en anillo.

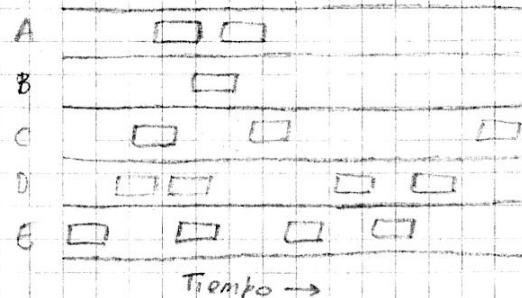
El desarrollo paulatino de distintos protocolos para el arbitraje del acceso al medio fue definiendo protocolos cada vez más confiables y con mejores prestaciones.

Topología en bus o lineal

El primer protocolo desarrollado para redes de difusión fue ALOHA. Implementado por primera

Vez en la universidad de Hawaii, permitía comunicarse rápidamente a un conjunto de computadoras distribuidas en distintas Islas.

Su idea es muy simple; dejar que las máquinas transmitieran como quisieran y detectar las colisiones retransmitiendo el paquete (ya que, debido a la propiedad de retroalimentación de la difusión, una máquina puede saber si su paquete colisionó con otro en el canal). Si la llama fue destrozada por una colisión, el transmisor espera un tiempo aleatorio (para evitar colisiones en sincronía) y lo envía nuevamente.

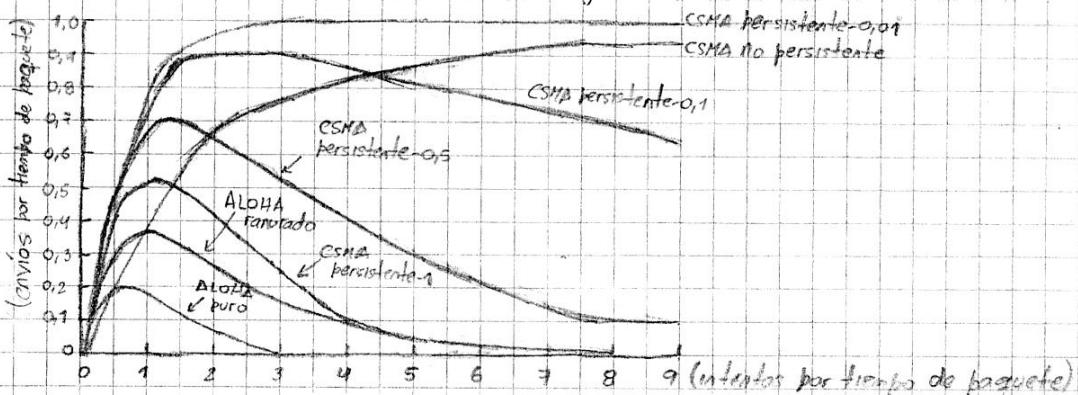


Este protocolo funcionó bien con baja cantidad de máquinas e información, pero con el crecimiento de la red debió buscarse una forma de evitar las colisiones (hay que tener en cuenta que incluso una colisión de un bit causa la destrucción de los dos tramas). Cuando hay muchas máquinas y muchos paquetes transmitiéndose, es más fácil que al segundo intento también se produzca una colisión, y las retransmisiones acastigantes aumentan esa posibilidad. Por lo tanto el rendimiento del protocolo es muy pobre, y empieza rápidamente a partir de cierta cantidad de tráfico.

Una extensión a este protocolo fue el ALOHA rizado, en el cual se transmitían paquetes sólo dentro de "ranuras" temporales sincronizadas por una estación que transmitía el comienzo y fin de estos períodos de tiempo. La mejora fue del doble; se pasó de un 18% del uso del canal a un 36%. De todas maneras, este protocolo no tiene un buen aprovechamiento del canal.

Para mejorar estas situaciones, es conveniente hacer que las estaciones detecten lo que hacen las demás estaciones y actúen en base a ello. Los protocolos en que las estaciones escuchan una por turnada (una transmisión) y actúan en consecuencia se llaman protocolos de detección de interferencia. Un ejemplo es el protocolo CSMA (Carrier Sense Multiple Access); antes de transmitir se detecta si el canal está en uso. Luego, tenemos varias posibilidades:

- Esperar a que el canal se desocupe y transmitir inmediatamente. Si hay una colisión espera un tiempo aleatorio y vuelve de nuevo. Se denomina persistente-1, pues al detectar el canal libre transmite con probabilidad 1. Se llega hasta un 50% de aprovechamiento del canal.
- Verificar el uso del canal en tiempos aleatorios hasta encontrar el canal en desuso (no persistente, menor aprovechamiento pero más retardos).
- Verificar el uso del canal en tiempos aleatorios y transmitir con probabilidad p al encontrarlo en desuso (persistente- p , hasta 95% de aprovechamiento según la probabilidad).



Otra mejora es que las estaciones aborten sus transmisiones tan pronto como detectan una colisión. Luego cada estación espera un tiempo aleatorio e intenta de nuevo. Este protocolo, conocido como CSMA/CD (CSMA with Collision Detection), aumenta notablemente el rendimiento del canal y es el que se usa actualmente en las redes tipo Ethernet.

Es importante notar que ninguno de estos protocolos garantiza la entrega confiable de la trama; incluso en la ausencia de colisiones, el receptor podría no haber recibido la trama por varias razones (ej: tener la memoria de entrada llena). Además, no se garantiza que la trama se transmitirá en un tiempo acotado; en teoría una trama podría demorar un tiempo arbitrariamente grande hasta poder ser transmitida, lo que es malo para sistemas de tiempo real.

Topología en anillo

Esta topología se basa en el concepto del paso de testigo. El testigo es una trama que se transmite entre las máquinas que conforman el anillo; cuando una de ellas desea transmitir información, toma el testigo, modifica su valor (toma posesión), y ninguna otra máquina podrá usarlo hasta que éste no sea liberado por la máquina receptora de la información. Cuento una estación posee el testigo, puede enviar información a otra estación destino; durante este tiempo todas las estaciones examinan el pequeño trama transmitido para verificar si son destinatarios del mismo, de no ser así se pasa el paquete.

te al verbo lógico, definido durante el inicio del sistema. En este tipo de protocolos se distinguen el modo de testigo en bus (Token Bus) y el modo de testigo en anillo (Token Ring).

IEEE 802.3 - Ethernet

Ethernet es el nombre dado a una tecnología LAN inventada por Xerox en los '70. IEEE libreró una versión compatible del estándar, que es diferente a Ethernet en varios puntos: especifica diferentes velocidades hasta 10 Mbps (Ethernet establece solo 10 Mbps), el campo de longitud en la trama en 802.3 se usa para el tipo de paquete en Ethernet, y otros detalles menores.

Cableado de Ethernet

- Ethernet de cable grueso (Thicknet o 10Base5): esta es la norma Ethernet original, y es hoy obsoleta pues ha sido reemplazada por alternativas más prácticas y económicas. Consiste en un cable coaxial de $\frac{1}{2}$ pulgada. La conexión requiere un conector tipo vampire. La distancia máxima es 500m, y en los extremos se debe añadir una resistencia para evitar la reflexión de señales. Se utiliza un transceptor (T/R) que se conecta físicamente el cable ether, y contiene la electrónica necesaria para manejar la detección de perturbación y de colisión, y para la comunicación en la tarjeta de interfaz en la PC. Además tiene un cable AUI (Attachment Unit Interface), que conecta al transceptor con la tarjeta del adaptador en la PC. Este cable transporta la potencia eléctrica para operar el transceptor, las señales de control, y la información de los tramos enviados y recibidos. Por último, hay una tarjeta interfaz, que contiene un chip controlador encargado de procesar los tramas, calcular y verificar las CRC, desearcar las tramas que no son para ella, y pasárselas la información transportada a la capa de enlace.

El nombre 10Base5 significa: 10 Mbps, modulación en banda base (no modulada) y 500m máximo por segmento.

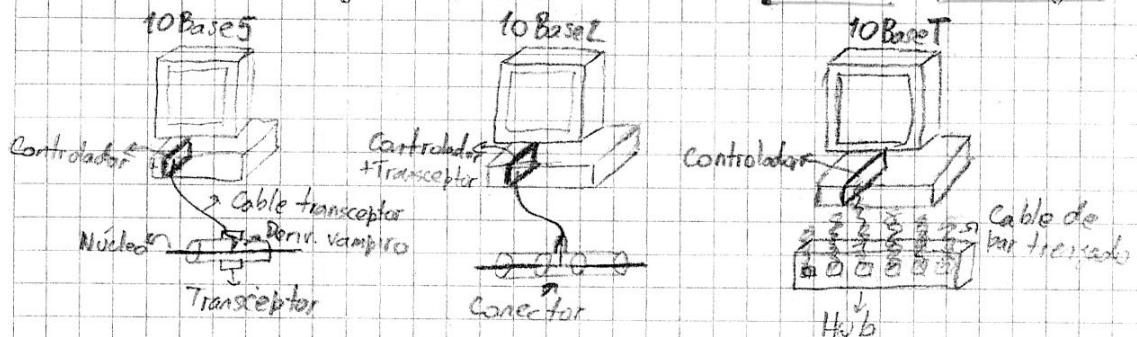
- Ethernet de cable delgado (Thinnet o 10Base2): esta configuración se creó porque el Thicknet en oficinas era difícil de colocar por el cable grueso y el costo de los transceptores. El cable coaxial es más delgado y la tarjeta interfaz se conecta directamente al cable para reducir los costos. Es ideal para redes de una habitación, si hace falta agregar una PC, simplemente se añadía a la cadena. Su conexión es muy simple de

bido a los conectores BNC y las fichas T. Una de las desventajas es que si se desconecta uno de los cables, queda incomunicada toda la red.

El nombre 10Base2 significa: 10 Mbps, modulación en banda base y zoom máximo por segmento.

- Ethernet de par trenzado (10BaseT): el avance en la tecnología permitió utilizar cable UTP; esta configuración es aún más económica y protege de una incompatibilidad por culpa de una PC. Cada computadora se conecta a un hub o concentrador, el cual conceptualmente es equivalente a juntar todo el bus en un punto. Físicamente es una caja que se puede dejar en un rack de cableado. La distancia entre una máquina y el hub no puede ser mayor a 100m. Los hubs más sofisticados pueden proporcionar a personal autorizado la capacidad de monitorear y controlar su operación. En efecto, es posible ver el tráfico en cada puerto, hacer estadísticas y aún habilitar y deshabilitar remotamente sus puertos.

El nombre 10BaseT significa: 10 Mbps, modulación en banda base, cable trenzado.



Se debe aclarar que la tecnología 10Base2 se utilizó mucho tiempo, cuando la electrónica era cara (los hubs costaban mucho). A partir de 1977/8 comprar un hub no era un gran gasto y su costo compensó crecer el no tener los problemas que da el cable coaxial. Hay otro costo que es el del cable: si bien el metro de cable UTP es más barato que el coaxial, se usan muchos más metros para la misma instalación. Esto se debe a que el coaxial es un único cable que pasa de máquina en máquina, mientras que usando UTP se requiere un cable individual entre cada máquina y el hub.

- Ethernet con fibra óptica (10BaseF): el método de conexión es igual al 10BaseT, solo que se conecta con fibras ópticas, a un hub con conectores para fibra. Esta opción tiene una excelente inmunidad a los ruidos. Si bien vienen hubs y switches con salidas Ethernet de fibra, la fibra se usa más que nada para comunicar segmentos que están alejados o separados.

por un ambiente hostil a los cables metálicos. Por ejemplo, si se deben comunicar dos edificios lo recomendable es usar fibra. En el mercado existen dispositivos llamados media converter que tienen una entrada UTP o BNC y otra de fibra, y permiten extender una red entre varios edificios. Hay hubs y switches que traen medias convertidoras incorporadas, de tal manera que se los pueda conectar a la red mediante éstos.

Nombre	Cable	Segmento máximo	Nodos por segmento	Ventaja
10Bases5	Coaxial grueso	500 m	100	Backbone
10Base2	Coaxial fino	200 m	30	Más barato
10BaseT	Par trenzado	100 m	104	Fácil de mantener
10BaseF	Fibra óptica	Variable	1024	Mejor en edificios

Propiedades de las redes Ethernet

La red Ethernet es una tecnología que se conoce como "entrega con el menor esfuerzo" y un control de acceso distribuido. Esto significa que el emisor no se enterá si las tramas llegan a destino o no, y que no tiene una autoridad central para garantizar el acceso al medio.

La eficiencia de Ethernet disminuye levemente respecto del número de estaciones en la red y aumenta con el número de bytes presentes en la trama.

Direccionalamiento Ethernet

Por ser una red de difusión, se debe utilizar un esquema de direcciones para permitir que las tramas sean procesadas sólo por los réceptores destinatarios. El direccionamiento Ethernet es en campo de 48 bits en la trama; cada tarjeta Ethernet tiene asociado un número único en el mundo.

La forma de asignar esta dirección a cada interfaz es mediante la administración por parte de la IEEE, que otorga a cada fabricante de placas un prefijo de direcciones de 3 bytes, y el fabricante asigna todas las direcciones que tienen ese prefijo a cada equipo que fabrica. Esta dirección se graba en una ROM.

Debido a que esta dirección está ligada con cada interfaz, se la llama dirección física. Si se cambia la interfaz, cambia la dirección física. Hay algunos números reservados para un uso especial.

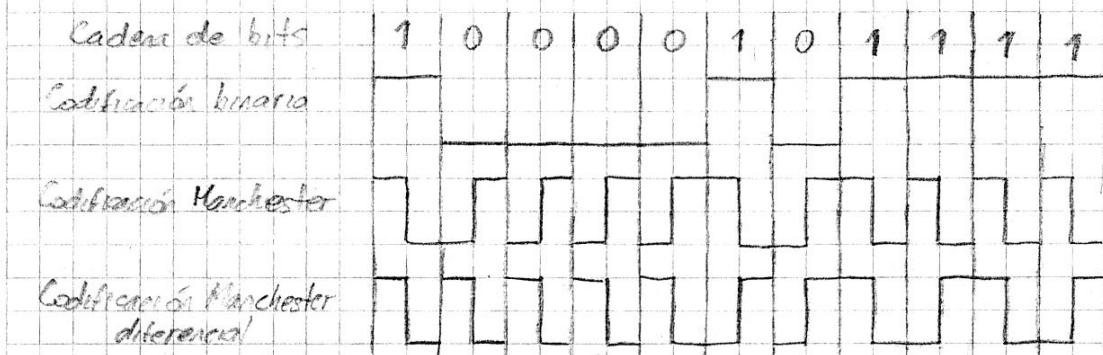
Esta dirección puede ser de tres tipos: una dirección física de interfaz de red (única difusión), una dirección de multi-difusión (grupos), o una dirección de difusión (todos), que consiste en todos los bits

en 1. La ventaja de estos dos últimos es que en conjunto de máquinas puede ser alcanzado con una sola trama. Para adaptarse a esto, la interfaz debe reconocer la dirección de (multi)difusión, además de su dirección física. Algunas interfaces pueden ser programadas para que acepten varias direcciones físicas (por ejemplo para multidifusión).

Transmisión en el medio

Hay dos formas básicas de transmitir la información en la banda base. Una es la codificación binaria, donde 0 voltios representan un 0 y 5 voltios significan un 1. El problema de esta codificación es que conduce a ambigüedades, y además las diferentes velocidades de reloj pueden causar que el emisor y el receptor pierdan la sincronía.

Por esto, se utiliza la codificación Manchester, donde se representa un 0 como una transición de un nivel bajo a alto en la línea, y un 1 como una transición de alto a bajo, asegurando de esta forma que en la mitad del tiempo de duración del dato existe una transición para permitir la sincronización de los receptores con el transmisor. Como desventaja, se requiere el doble de ancho de banda. Una variante es la codificación Manchester diferencial, donde un 0 se representa mediante la presencia de una transición al comienzo del intervalo, y un 1 como la ausencia de transición. En ambos casos hay una transición a la mitad. Este esquema requiere equipo más complejo, pero ofrece mayor inmunidad al ruido. Todos los sistemas Ethernet usan la codificación Manchester.



Formato de la trama Ethernet

La trama utilizada es de una longitud mayor a 64 bytes y menor a 1518 bytes.

7	1	6	6	2	0 - 1500	0 - 46	4
Preámbulo	Inicio	Dir. destino	Dir. origen	Largo	Datos	Relleno	CRC

- Preámbulo: combinación de bits que permite la sincronización del transmisor con los receptores, ésta luego se mantiene gracias a la codificación Manchester.

- Inicio: un octeto de inicio donde converge la información (10 10 10 11).

- Dir. destino: dirección física a la cual se dirige la trama.
- Dir. Origen: dirección física donde se originó la trama.
- Larg. tuc de datos: contiene la cantidad de bytes que ocupa el campo de datos (en Ethernet original se usa para el tipo de dato transportado) ($<0x600$ longitud, $>0x600$ tipo).
- Datos: los datos transportados.
- Relleno (Pad): utilizados cuando la cantidad de datos no cubre el mínimo requerido.
- CRC: verificación de redundancia cíclica.

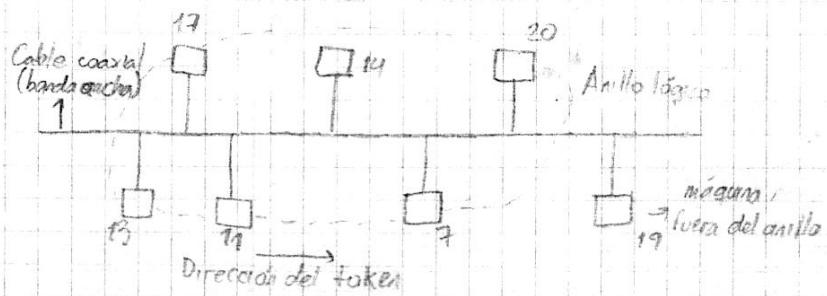
Si se recibiera una trama menor a 64 bytes, se la descarta por la posibilidad de que sea restos de una trama colisionada.

Mejoras a Ethernet

Posteriormente, se creó Fast Ethernet, con una velocidad de 100 Mbps, y Gigabit Ethernet, con una velocidad de 1 Gbps. En ambos casos, en la actualidad se los puede transmitir por UTP o fibra óptica (no por coaxial, que sólo funciona a 10Mbps).

IEEE 802.4-Token Bus

Aunque Ethernet (IEEE 802.3) de 10 Mbps está muy difundido en redes de oficinas, la característica aleatoria de su protocolo MAC (que en el peor de los casos haría a una estación esperar mucho tiempo para transmitir) y la imposibilidad de definir prioridades de transmisión no lo hacen adecuado para uso industrial, especialmente en sistemas de control, donde la posibilidad de definir prioridades a distintas estaciones es relevante. Si es de utilidad el uso de un canal de difusión lineal o en bus. Estas razones llevaron a implementar una nueva norma basada en paso del testigo, que implementa en anillo lógico sobre una topología física lineal.



En este caso se utiliza para el medio un cable coaxial de banda ancha de 75 ohms, la señal es modulada de forma similar al FM, con velocidades de 10Mbps y una longitud

del cable no mayor a 100m.

El funcionamiento corresponde al detallado anteriormente como paso del testigo, sólo que el mismo se propaga de estación en estación, recorriendo un anillo lógico; como sólo una estación puede tener el testigo, no existe posibilidad de colisiones.

Cuando se inicia la red, las estaciones van entrando según su dirección, de mayor a menor, este orden también determina el orden del paso del testigo.

A diferencia del CSMA/CD, en este caso todas las estaciones escuchan, pero sólo transmitir cuando el orden lógico del testigo lo permite. La posesión del testigo está limitada en el tiempo, y a su vez cada estación determina cuatro niveles de prioridad decreciente para la transmisión de sus datos, asignando secciones del tiempo directamente proporcionales al orden de prioridad. En caso de que el tiempo no sea suficiente, la estación deberá esperar hasta tener el testigo nuevamente.

Deben manejarse dos tipos de tramas: las de control y las de datos. Cuando una estación modifica el valor del campo de control utilizando que ha tomado la posesión del testigo, puede transmitir un determinado número de tramas durante el tiempo asignado a la posesión del testigo, luego lo libera y se pasa a la estación siguiente según el orden establecido. El formato de la trama es similar a Ethernet, incluyéndose los campos de dirección.

IEEE 802.5 - Token Ring

En realidad esto no es un protocolo MAC, pues opera físicamente punto a punto. Los tramas se transmiten bit a bit entre las máquinas adyacentes (la diferencia de los dos anteriores, donde una máquina transmite y todas escuchan).

Para la capa física se especifica UTP categoría 5, operando de 4 a 6 Mbps, con codificación Manchester. El perímetro no debe ser mayor a 360m.

Un problema es que si se rompe una conexión el anillo desaparece. Esto se soluciona utilizando un dispositivo que concentre todas las conexiones en un centro de estrella física, llamado MAU (Multiple Access Unit).

El funcionamiento básico es simple: cuando no hay tráfico circula el testigo por el anillo, cada máquina almacena cada bit temporalmente y lo retransmite a la máquina adyacente, operando en el modo para escuchar. Cuando una de las máquinas requiere del testigo

modifica un bit específico del mismo o interrumpe la transmisión; cuando este bit modificando es detectado por las otras máquinas del anillo, éstas van entrando ordenadamente en modo de transmisión-recepción, adquiriendo la totalidad de la trama para procesarla, ya que ésta presenta también un campo de dirección.

Comparación entre 802.3, 802.4 y 802.5

Ethernet es simple en instalación y operación del protocolo MAC. No es necesario utilizar nodos y los costos son reducidos. Es eficiente con tramas de datos largos, pero si aumenta el tráfico, aumentan las colisiones y la red es menos segura. Para un número medio de 250 estaciones con tráfico moderado resulta bastante eficiente.

Token Bus es más elaborado y requiere equipo más sofisticado de transmisión. Se orienta a la industria, aunque se usa en oficinas cuando el tráfico es elevado y la seguridad es primordial, y fundamentalmente cuando se quieren gestionar prioridades de atención.

Token Ring es simple y económico, pero su inconveniente está en su funcionamiento, ya que depende del buen estado y la velocidad de las estaciones. Si una estación sale de servicio, se cae toda la red a menos que se use un RAU, que cerraría el anillo, pero su costo es elevado. Resume las mismas características que Token Bus, pero es solamente de uso comercial, ya que usa banda base como transmisión.

La economía de escala (con la consiguiente reducción de precios) hizo que 802.4 y 802.5 desaparecieran de las redes de oficinas, utilizando casi con exclusividad 802.3.

Dispositivos para extender las redes 802

Existen varios dispositivos diseñados para extender y mejorar el funcionamiento de las redes físicas; estos son repetidores y puentes (más recientemente switches).

Repetidores

Conceptualmente, este tipo de dispositivos corresponde a la capa 1. Consiste simplemente en un amplificador de señales eléctricas. No interpreta los datos de la trama. En la práctica se usan como parte integrante de los hubs (permitten retransmisión) cuando se debe pasar de un medio a otro. Los media converter mencionados anteriormente son repetidores que tienen puertos

que se conectan a distintos tipos de medio físico. Un repetidor sólo sirve para conectar un tipo de red (es decir, no se puede usar para pasar, por ejemplo, de 802.3 a 802.5).

Puentes

Los puentes operan en capa 2, lo cual quiere decir que están en condiciones de interceptar las tramas y modificarlas. Los puentes tienen 3 usos principales.

Reducción de la saturación en una red

Esto se da cuando la cantidad de estaciones y/o el tráfico aumentan al punto de que se produzcan muchas colisiones. Se divide la red en dos segmentos y se conecta cada uno a un puerto del puente. Los puentes utilizados son inteligentes: están continuamente escuchando el tráfico en ambos segmentos y, en base a las direcciones de origen que llevan las tramas, se arma una tabla donde figuran las direcciones MAC de cada máquina y su segmento correspondiente. La mejora en el rendimiento se logra cuando el puente ya conoce esta información. Entonces, el tráfico cruza de segmento sólo cuando es necesario, disminuyendo así la saturación de la red. Cuando un puente ve una trama dirigida a una dirección que aún no tiene en sus tablas, retransmite la misma al otro segmento. Tardé o temprano la máquina llamada contestará y entonces el puente tomará nota de su ubicación.

En la actualidad, la función del puente está incluida en unos dispositivos llamados switches, que consisten en puentes con muchos puertos. A un switch se pueden conectar hubs con varios módulos cada uno, para disminuir las colisiones y aumentar el rendimiento de la red. En las redes modernas, esto puede llevar al extremo de conectar sólo una máquina a cada puerto del switch, y con esto lograr una red Ethernet sin colisiones. Últimamente se ha ido consiguiendo switches muy sofisticados, que pueden operar y tomar decisiones en capas 3 o 4.

Actualmente se ofrecen en el mercado hubs totales, es decir, que soportan ambas velocidades. Cuando se quiere pasar una trama que entró por un puerto de 100 Mbps a un puerto de 10 Mbps, se debe almacenarla y retransmitirla a la velocidad correspondiente. En efecto, estos equipos tienen en su interior un puente con dos segmentos. Todos los puertos de 10 Mbps se conectan al segmento de baja velocidad, mientras que los de 100 se conectan al de alta velocidad.

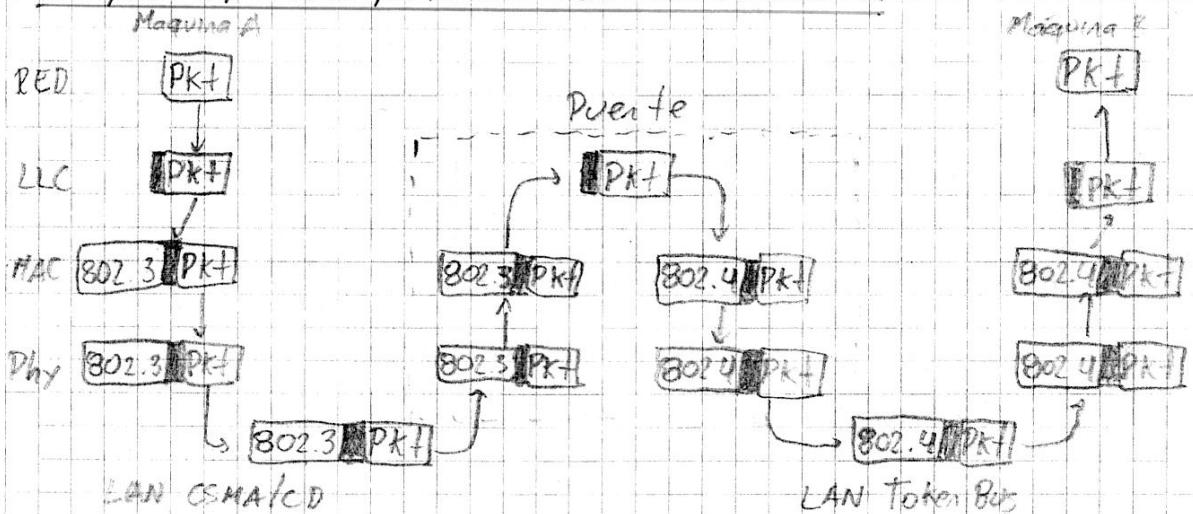
Conexión de segmentos alejados

Cuando se da este caso, el retraso puede causar graves problemas en los protocolos.

los tipo CSMA/CD. Para solucionar esto se colocan puentes en los extremos del enlace de tal forma de recibir, almacenar y retransmitir las tramas cuando sea apropiado.

Comunicación entre tecnologías diferentes

En estos casos no es suficiente un repetidor, ya que se deben interpretar y modificar las tramas para adaptarlas al protocolo de enlace del otro extremo.



Se podría pensar que es sencillo transformar tramas en una 802, pero no es así. Cada una de las combinaciones tiene su propio grupo de problemas. Algunos (generales) son:

- Cada una de las LAN utiliza un formato de trama diferente. Esto se debe a que las empresas que promovieron cada estándar no querían cambiar el suyo cuando la IEEE los tomó. Esto implica que cualquier copiado entre diferentes LANs requiere reformateo, lo que exige tiempo de CPU, obliga a un nuevo cálculo de suma de comprobación e introduce la posibilidad de errores no detectados debido a bits erróneos en la memoria del puente.

- Las LANs interconectadas no necesariamente operan con la misma tasa de datos. Al enviar un grupo grande de tramas una tras otra de una LAN rápida a una lenta, el puente no será capaz de deshacerse de los marcos a la misma velocidad a la que llegan; tendrá que encolarlos en memoria y esperar que ésta no se acabe.

- Las tres LAN 802 poseen una longitud máxima de trama diferente. Esto se convierte en un problema sin solución, ya que a este nivel no se puede realizar fragmentación de tramas; las LAN 802 no lo soportan. En estos casos, las tramas que son de mayor longitud que el máximo permitido por la LAN de destino se descartan.

Redes inalámbricas

Una red de área local inalámbrica (WLAN) es una red que cubre un área equivalente a la red local de una empresa, con un alcance aproximado de 100 metros. Permite que los nodos que se encuentran dentro del área de cobertura puedan conectarse. El medio de comunicación es a través de ondas electromagnéticas (radio e infrarrojo) en lugar de cableado estandar. La banda de frecuencias ICH utilizada es de uso libre siempre que se respeten los límites de la potencia irradiada. La variedad en las tecnologías radica en la frecuencia de transmisión, el alcance y la velocidad que utilizan. Se debe tener en cuenta que las frecuencias utilizadas no pueden ser asignadas en forma aleatoria por el fabricante porque existen regulaciones legales en cada país para controlar el uso eficiente y adecuado del espectro electromagnético como así también la potencia de transmisión irradiada.

Actualmente podemos diferenciar 3 tipos de redes inalámbricas:

- WPAN: son redes pensadas para cubrir una habitación. Tradicionalmente, los infrarrojos dominaron este tipo de red. Actualmente, la tecnología Bluetooth es el estándar en este tipo de red.
- WLAN: son redes que cubren el ámbito de una casa, una oficina, o un edificio.
- WWAN: son redes que cubren áreas más amplias, como una ciudad. El desarrollo actual está dado principalmente por las empresas de telefonía móvil.

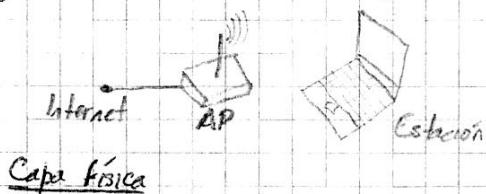
Wi-Fi - 802.11

El protocolo IEEE 802.11 o Wi-Fi es un estándar de protocolo de comunicaciones de la IEEE que define el uso de los dos niveles más bajos de la arquitectura OSI, especificando sus normas de funcionamiento en una WLAN. El estándar original de este protocolo data de 1997, tenía velocidades de 1 a 2 Mbps y trabajaba en 2,4 GHz (en la actualidad está en desuso). La siguiente modificación apareció en 1999 y fue designada como IEEE 802.11b; tenía velocidades de 5 a 11 Mbps y continuaba trabajando en 2,4 GHz. Se realizó una especificación, 802.11a, para 5 GHz que alcanzaba los 54 Mbps pero resultaba incompatible con 802.11b. El estándar que incorpora esa velocidad y es compatible es el 802.11g. La seguridad forma parte del protocolo desde el principio y fue mejorada en la revisión 802.11i. El estándar 802.11n mejora significativamente el rendimiento de la red, con un incremento significativo en la velocidad máxima de transmisión de hasta 600 Mbps. La industria ya está trabajando en nuevos productos

y dispositivos basados en el protocolo 802.11ac, que permite velocidades de al menos 1000 Mbps en la banda de los 5 GHz. Además, el alcance de cobertura es ampliamente superior a otras versiones, de modo que llega hasta un máximo de 90-100 metros.

• Arquitectura

Una red LAN inalámbrica 802.11 está basada en una arquitectura celular donde el sistema se subdivide en celdas. Cada celda (llamada Basic Service Set) se controla por una estación base (Access Point). El sistema puede estar formado por una única celda con un único AP, o por varias celdas donde los APs están unidos por un tipo de enlace troncal, siendo una red Ethernet la más utilizada. Este sistema en su conjunto se llama Distribution System.



• Capa física

El estándar 802.11 ha tenido una evolución constante desde su nacimiento, es por esto que existen varias versiones, cada una con especificaciones y mejoras diferentes.

- 802.11 Legacy: fue creado en 1997 y es considerado como la versión original. Permite velocidades de transmisión de hasta 2 Mbps y opera sobre la frecuencia de 2,4 GHz. Utiliza el protocolo de acceso al medio CSMA/CA. Este estándar tiene dificultades de comunicación entre dispositivos de diferentes marcas y problemas de interferencia al utilizar la misma banda de frecuencia que electrodomésticos.

- 802.11a OFDM: aprobado en 1999. Permite realizar transmisiones con velocidades de hasta 54 Mbps y opera sobre la banda de frecuencias de 5 GHz. El alcance aproximado es de 25 metros. Utiliza 52 subportadoras mediante la multiplexación por división de frecuencias OFDM.

- 802.11b DSSS: ratificado en 1999. Soporta velocidades de hasta 11 Mbps y opera sobre la banda de Frecuencias 2,4-2,5 GHz. El alcance aproximado es de 50 metros. Es el más popular pues fue el primero en imponerse y existe un inventario muy grande de equipos y dispositivos que manejan esta tecnología. Al igual que la versión original, utiliza CSMA/CA. Usa la técnica de Espectro Enanchado por Secuencia Directa(DSS) para la modulación de la señal, enviando también bits de redundancia que evitan retransmisiones y corrigen los errores de la framza.

- 802.11g: ratificado en junio de 2003. Opera a una velocidad de transmisión máxima de 54 Mbps y emplea las mismas bandas de frecuencia que 802.11b, haciéndolo compatible; aunque combinar ambos estándares reduce significativamente la velocidad de transmisión. Emplea las técnicas de modulación OFDM y DSSS. Con potencias de hasta medio watt y antenas parabólicas apropiadas, puede llegar a hacer comunicaciones de hasta 50 km.

- 802.11n MIMO: ratificado en septiembre de 2009. Opera a una velocidad de transmisión máxima de 600 Mbps, aunque la capa física actualmente soporta 300 Mbps con el uso de dos flujos espaciales en un canal de 40 MHz. Cuenta con una velocidad de modulación cerca de 6 veces más rápida y una tasa de transferencia de datos de 2 a 5 veces mayor que una antena Wi-Fi 802.11a/g. En este nuevo estándar se utiliza un OFDM mejorado, que provee anchos de banda más amplios y mayores velocidades de datos. Además se implementa agregación de tramas, que reduce la transmisión de encabezados ya que permite que varios tramas de datos se envíen como una sola transmisión.

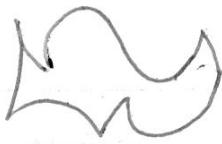
802.11ac MU-MIMO: el nuevo estándar opera en el espectro de 5GHz, frecuencia que restringe la distancia de transmisión, pero que otorga menor sensibilidad a los obstáculos físicos, ademas de estar menos saturada. Este estándar amplia los flujos espaciales hasta un máximo de 8, aunque probablemente las primeras implementaciones llegan hasta 4. Además, se espera que 802.11ac sea energéticamente más eficiente.

Espacio disperso con salto de frecuencia (FHSS)

Esta tecnología consiste en transmitir una parte de la información en una determinada frecuencia durante un intervalo de tiempo llamado tiempo de permanencia (dwell time). Pasado este tiempo se cambia la frecuencia de emisión y se sigue transmitiendo en otra frecuencia. De esta manera cada tramo de información se va transmitiendo en una frecuencia distinta durante un intervalo muy corto de tiempo. El orden en los saltos de frecuencia se determina según una secuencia pseudoaleatoria que el emisor y el receptor deben conocer.

Si se mantiene la sincronización en los saltos de frecuencia se consigue que, aunque en el tiempo se cambie de canal físico, a nivel lógico se mantenga un solo canal por el que se realiza la comunicación.

Para la banda de 2.4 GHz, el estándar 802.11 define el orden de los saltos en 3 conjuntos, con 26 secuencias cada uno. Las secuencias cubren 79 canales a lo largo de la banda, con secuencias independientes unas de otras en cada conjunto.



Especro disperso con secuencia directa (SSS)

Esta tecnología consiste en modulir la señal a transmitir con una secuencia de bits de alta velocidad, denominados en este caso chips y conocida como secuencia de Barker, código de dispersión o ruido pseudo-aleatorio (código PN), que da como resultado una expansión de la señal.

Si bien la relación potencia-ancho de banda se mantiene después del ensanchamiento de la señal, se obtiene una señal más inmune al ruido, ya que la interferencia afecta sólo unos bits de la señal original. La secuencia de Barker recomendable es de 11 chips, aunque puede llegar a 100, pero a mayor número de chips, mayor es el costo de los osciladores de radiofrecuencia necesarios para su manejo y mayor es el ancho de banda requerido.

Múltiple entrada/múltiple salida (MIMO)

MIMO aprovecha la técnica de diversidad espacial para el extremo receptor y transmisor; todo esto tiene por objeto mejorar la fiabilidad y calidad del enlace, reduciendo la tasa de error en bit; esta reducción permite el uso de esquemas de modulación superiores y mejora la eficiencia espectral.

Como los sistemas MIMO hacen uso de múltiples antenas en transmisión y recepción, esto da lugar a la generación de subcanales equivalentes paralelos e independientes, que a su vez generan una ganancia que viene dada por los valores propios obteniéndose una matriz $M \times N$, donde M es la antena y N el subcanal generado por dicha antena.

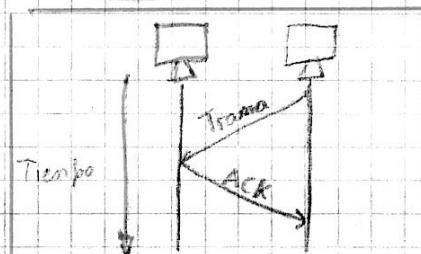
El rendimiento y la mejor eficiencia del canal MIMO se mide a través de la capacidad del canal, a partir de las expresiones de capacidad propuestas por Shannon para canales simples; la capacidad del canal MIMO solo defenderá de la relación señal-ruido en el receptor y de la matriz de canal, independientemente del esquema de transmisión o codificación utilizado.

Para la transmisión tenemos el multiplexado espacial que aprovecha el dominio espacial para obtener altas tasas de transferencia y por otra parte la codificación espacio-temporal que está más orientada a mejorar la fiabilidad y calidad del enlace, mediante el uso de la diversidad espacial desde otra concepción, que establece la transmisión de múltiples réplicas de cada símbolo.

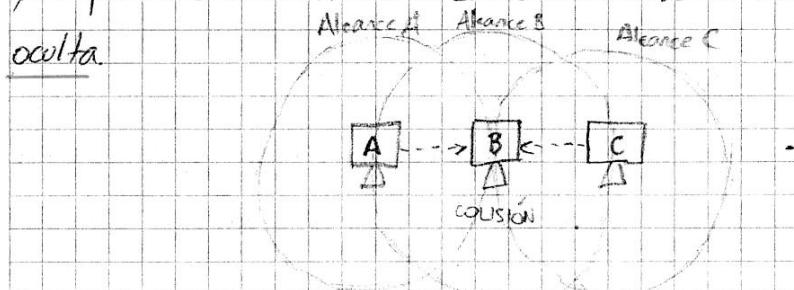
Capa de enlace

Ethernet transmite la trama y asume que el destino lo recibe correctamente. En los enlaces de

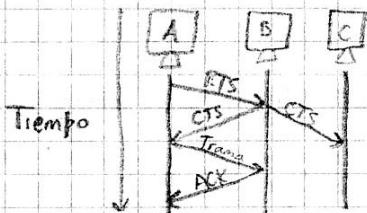
radio esto es diferente, especialmente cuando las frecuencias usadas pertenecen a los bandos de ICM, que están sujetas al ruido y a la interferencia. Es por esto que toda transmisión en 802.11 debe ser reconocida a través de un aviso de recepción (ACK): si cualquiera de los partes falla, la trama se considera perdida.



Podría ocurrir que dos estaciones no puedan comunicarse directamente, porque el alcance es insuficiente y que se produzca una colisión no detectada por ellas, que transmiten al mismo tiempo, y sí por una estación intermedia. Esta situación se conoce como problema de la estación oculta.



IEEE 802.11 implementa señales de requerimiento y aceptación para el envío de datos con el propósito de reservar el enlace de radio (request to send y clear to send) y prevenir colisiones. Este proceso sucede de la siguiente manera: A envía una trama RTS a la estación B, ésta envía una trama CTS como respuesta a las estaciones A y C, la estación C se detiene, y no transmite durante el tiempo que dure la transmisión A envía la trama de datos a B y finalmente, B envía la trama de ACK a la estación A.



Modos de acceso

El acceso al medio inalámbrico es controlado por funciones de coordinación:

- DCF (Función de Coordinación Distribuida): la DCF es el mecanismo de acceso básico del estándar 802.11, donde primero se verifica que el enlace de radio se encuentre limpio para transmitir. Para evitar colisiones, las estaciones retardan aleatoriamente el envío de tramas y luego escuchan el canal para poder transmitir. En algunas circunstancias, la DCF puede usar la técnica de RTS/CTS para reducir la posiblidad de colisiones.

bilidad de colisiones.

-PCF (Función de Coordinación Puntual): esta función está asociada a transmisiones libres de contienda, las cuales utilizan técnicas de acceso deterministas. Para esta función se define una técnica de interrogación circular desde el punto de acceso. Esta funcionalidad es para servicios de tipo síncrono que no soportan retrasos aleatorios en el acceso al medio.

El problema de la estación expuesta ocurre cuando una estación B transmite a una estación A, y otra estación C, cercana a B, quiere transmitir a una estación D. C detectará la transmisión de B y decidirá no transmitir, pese a que no hay interferencia. Esto también se soluciona con el esquema RTS/CTS.

Dos
ra por
tación

del medio a las capas superiores.

CSMA: Iro veo el medio \rightarrow dps transmito

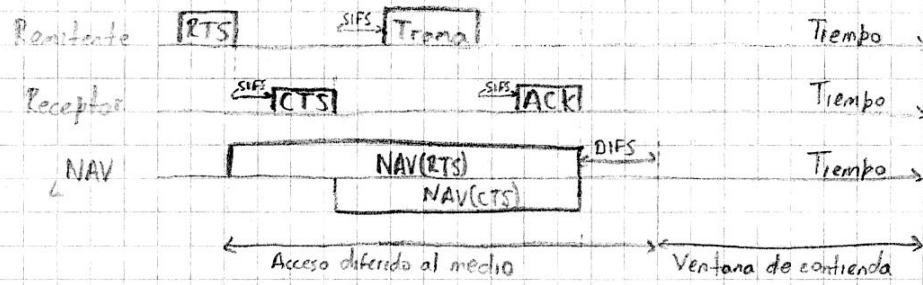
Las funciones de detección de portadora son provistas por la capa física dependiendo del medio y de la modulación utilizada.

Las funciones de detección de portadoras virtuales son provistas por el Vector de Asignación de Red (NAV). El NAV es un temporizador que indica la cantidad de tiempo que el medio será reservado, expresado en microsegundos. La estación coloca el NAV con el tiempo que espera ocupar el medio, incluyendo otras tramas necesarias para completar la operación. Las otras estaciones realizan una cuenta regresiva desde el NAV hasta llegar a 0. Cuando el NAV es distinto de 0, la función de detección de portadora virtual indica que el medio está ocupado y cuando llega a 0, que está disponible.

Cuando una estación está lista para transmitir, primero envía una solicitud de RTS al AP, la cual contiene el destino y la longitud del mensaje. El AP difunde el NAV a todos los demás nodos para que todos queden informados que se va a ocupar el canal y cuál va a ser la duración de la transmisión. Los nodos dejarán de transmitir durante el tiempo indicado por el NAV más un intervalo extra aleatorio (backoff). Si no encuentra problemas, el AP responde con una autorización (CTS) que permite al solicitante enviar su trama de datos. Si no se recibe la trama CTS, se supone que ocurrió una colisión y los procesos RTS empiezan nuevamente. Luego de recibida la trama de datos se devuelve una trama ACK no cifrando al transmisor que se ha recibido correctamente la información.

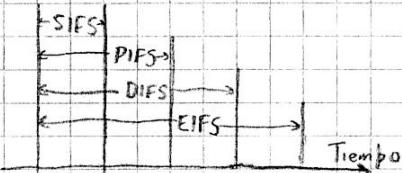
Un inconveniente que sigue presente radica en las colisiones producidas cuando varios estaciones envían

tramas RTS al mismo tiempo, aunque estas colisiones son menos dañinas al ser su tiempo de duración relativamente corto.



Espaciamiento intertrama

El estándar 802.11 usa cuatro diferentes espaciamientos entre tramas. Variando este espaciamiento se pueden crear diferentes niveles de prioridad para distintos tipos de tráficos. Debido a las distintas capas físicas que el estándar 802.11 puede adoptar, éstas pueden especificar diferentes tiempos de intertrama.



-SIFS (espaciado corto entre tramas): es usado para transmisiones de alta prioridad, tales como RTS, CTS y ACK positivas.

-PIFS (espaciado entre tramas PCF): es usado por la PCF durante una operación libre de contienda. Las estaciones con datos para transmitir pueden hacerlo luego del PIFS.

-DIFS (espaciado entre tramas DCF): es el tiempo mínimo para servicios basados en contienda en el cual el medio debe estar libre para que una estación pueda acceder. Las estaciones pueden tener acceso inmediato al medio si ha estado libre por un tiempo mayor al DIFS.

-EIFS (espaciado extendido entre tramas): es usado solamente cuando hay un error en la transmisión de una trama.

Estructura de la trama 802.11

Existen tres tipos de trama:

-Tramas de datos: usadas para la transmisión de datos.

-Tramas de control: usadas para el control de acceso al medio (RTS, CTS, ACK, etc.).

-Tramas de gestión: son trasmisibles de la misma manera que los tramas de datos para intercambiar información de administración, pero no son transportadas a las capas superiores.

Trama Wi-Fi

Hoja 29. 0.23.17 4

Bytes	2	2	6	6	6	2	6				
Bits	2	2	4	1	1	1	1	1			
	Versión	Tipo	Subtipo	A DS	De DS	MF	Retrans.	Ener.	Más	W	O

→ Control de Trama

- Control de trama: corresponde al comienzo de la trama compuesto por 2 bytes:

- Versión de protocolo: estos 2 bits indican cual es la versión de 802.11 MAC que se encuentra contenida en la trama. Hasta el presente existe una sola y se coloca un 0.
- Tipo: este campo indica el tipo de trama utilizado: gestión, control o datos.
- Subtipo: en este campo se detallan las acciones de cada uno de los tramas: asociación, reasociación, prueba, Beacon, disociación, autenticación, desautenticación, RTS, CTS, ACK, datos, etc.
- A DS y De DS: estos bits indican si una trama es destinada al sistema de distribución. Todas las tramas sobre la red de infraestructuras tienen estos bits activados.
- More Fragments (MF): este bit indica si la trama sufre alguna fragmentación. Mientras no sea el último fragmento, este bit está en 1.
- Retransmisión: cualquier trama retransmitida tiene este bit en 1 para excluir a la estación que la recibe en la eliminación de tramas duplicadas.
- Administración de la energía: para conservar la vida de la batería, muchos dispositivos tienen la capacidad para pasar la energización a la parte de la interfase de red. Un 1 indica que la estación entró en modo de ahorro de energía. Los AP poseen una importante función de administración y no pueden pasar al modo de ahorro de energía.
- Más datos: este bit es usado por el AP para indicar que hay más tramas para esa estación.
- Privacidad Inalámbrica Equivalente (WEP o W): si la trama está protegida por un protocolo de seguridad en la capa de enlace este bit está activado.
- Orden (O): este bit activado indica que las tramas y fragmentos van a ser transmitidos en un estricto orden. Esto produce un incremento del procesamiento durante el envío y recepción.
- Duración: este campo tiene dos significados dependientes del tipo de trama: en modo ahorro de energía corresponden de al 10 de la estación y en el resto indica la duración calculada que utiliza el NAV.
- Dirección 1: es siempre la dirección del receptor. Si A DS está activado representa la dirección del AP

y desactivado es la dirección de la estación.

-Dirección 2: es siempre la dirección del transmisor. Si De DS está activado representa la dirección del AP y desactivado es la dirección de la estación.

-Dirección 3: cuando el bit de De DS está activo representa la dirección de la fuente de origen. Si el bit de A DS está activo y no interesa el valor de De DS, representa la dirección de destino.

-Control de secuencias(Seq): este campo es usado para representar el orden de diferentes fragmentos dentro de una misma trama y para el reconocimiento de paquetes duplicados. Está constituido por el número de fragmento y el número de secuencia.

-Dirección 4: es usado en casos especiales cuando un sistema de distribución inalámbrico es usado, y las tramas son transmitidas desde un AP a otro. Los bits de A DS y De DS están activados en este caso.

-Suma de verificación: es un campo conteniendo un CRC para la detección de errores en la trama.

Control de enlace lógico(LLC)

Especifica los mecanismos para el direccionamiento de estaciones conectadas al medio y para controlar el cambio de datos entre usuarios de la red. La operación y formato de este estándar está basado en el protocolo HDLC. Establece tres tipos de servicio: sin conexión y sin reconocimiento, con conexión y sin conexión y con reconocimiento.

Redes WPAN

Se usan generalmente para conectar dispositivos periféricos o un PDA a un ordenador sin conexión por cables. También se pueden conectar dos ordenadores cercanos. La tecnología que se emplea en estas redes procura hacer eficiente el uso de recursos. Para lograr tal fin se diseñaron protocolos simples y los más óptimos para cada necesidad de comunicación y aplicación.

-Bluetooth: lanzado por Ericsson en 1994. Ofrece una velocidad máxima de 1 Mbps. Esta tecnología tiene la ventaja de tener un bajo consumo de energía.

-HomeRF: lanzada en 1998 por HomeRF Working Group. Ofrece una velocidad máxima de 10 Mbps con un alcance de 50 a 100 metros. Este estándar se abandonó al surgir WiFi en placa.

-Zigbee: se puede utilizar para conectar dispositivos en forma inalámbrica a un muy bajo costo y con bajo consumo de energía. Su principal aplicación está en pequeños aparatos electrónicos. Puede alcanzar una velocidad máxima de 250 Kbps con un alcance de 100 metros.

-Infrarrojo: este tipo de conexiones se puede utilizar en radios de muy pocos metros, con velocidad

des que pueden alcanzar algunos Mbps. Esta tecnología se usa ampliamente en aparatos electrónicos del hogar, pero puede sufrir interferencias debido a las ondas de luz.

El estándar que rige las redes WPAN es el IEEE 802.15. En base a la tasa de transferencia de datos, consumo de energía y QoS se han definido 4 clases:

- 802.15.1: se adoptó Bluetooth como base del estándar.
- 802.15.2: definición de modelos de coexistencia entre redes WPAN y WLAN.
- 802.15.3: estándar WPAN de alta velocidad para servicios multimedia.
- 802.15.4: estándar WPAN de baja velocidad, baja complejidad y bajo costo.

Bluetooth

La tecnología Bluetooth comprende hardware, software y requerimientos de interoperabilidad, por lo que su desarrollo ha necesitado la participación de los principales fabricantes de los sectores de las telecomunicaciones y la informática. Entre sus principales características pueden nombrarse su robustez, baja complejidad, consumo de energía y costo.

Funcionamiento

Cada dispositivo está equipado con un microchip llamado transceptor, que transmite y recibe en la frecuencia de 2,4 GHz, y tiene una dirección única de 48 bits. Estos dispositivos se clasifican en 3 clases en referencia a su potencia de transmisión:

- Clase 1: tiene una potencia máxima de 100 mW y un alcance de 100 m.
- Clase 2: tiene una potencia máxima de 2,5 mW y un alcance de 10 m.
- Clase 3: tiene una potencia máxima de 1 mW y un alcance de 1 m.

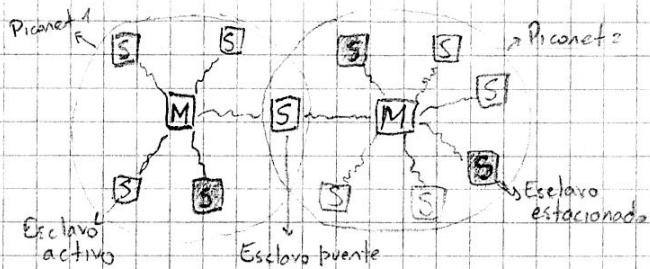
Debido a la naturaleza de las transmisiones, los dispositivos no deben estar alineados. Por lo general, el receptor será de clase 1, mientras que el emisor será de clase 3. De esta manera, la mayor sensibilidad y potencia del receptor permite que la señal llegue con energía suficiente hasta el emisor. Además, el receptor reconocerá la señal del emisor aunque sea débil.

Arquitectura

La unidad básica de un sistema Bluetooth es una node; que consta de un nodo maestro y hasta 7 nodos esclavos activos a una distancia de 10 metros. En una misma sala se pueden encontrar

trar varias piconets y se pueden conectar mediante un nodo puente. Un conjunto de piconets interconectadas se denomina scatternet.

Además de los 7 nodos esclavos activos de una piconet, puede haber hasta 255 nodos esclavos estacionados. Estos son dispositivos que el nodo maestro ha cambiado a un estado de bajo consumo de energía. Lo único que un dispositivo en este modo esclavizado puede hacer es responder a una señal de activación por parte del maestro. También existen los estados intermedios hold y sniff.



Protocolos

El estándar Bluetooth cuenta con muchos protocolos agrupados en poco orden en capas. Esta estructura no sigue el modelo OSI, el modelo TCP/IP, el modelo 802 o algún otro modelo conocido.

Aplicaciones / perfiles						
Audio	Otros	RFComm	Telefonía	Descubrimiento de servicios	Control	
	LLC	Protocolo de adaptación y control de enlaces lógicos		Administrador de enlaces		
Banda base						
Radio física						

Capa de radio física

La capa de radio traslada los bits del maestro al esclavo, y viceversa. Es un sistema de baja potencia con un rango de 10 metros que opera en la banda de 2,4 GHz. La banda se divide en 79 canales de 1 MHz cada uno. La modulación es por desplazamiento de frecuencia, con 1 bit por Hz, lo cual da una tasa de datos aproximada de 1 Mbps. Para asignar los canales de manera equitativa, el espectro de saltos de frecuencia se utiliza a 1600 saltos por segundo, y un tiempo de permanencia de 625 µseg. Todos los nodos de una piconet saltan de manera simultánea, y el maestro establece la secuencia de salto.

Debido a que tanto el 802.11 como el Bluetooth operan en los mismos 79 canales, interfieren entre sí, pero como el Bluetooth salta mucho más rápido, es más probable que éste dane las

transmisiones del 802.11 que el caso contrario.

Capa de banda base

Esta capa es lo más parecido a una subcapa MAC, ya que convierte el flujo de bits puros en tramas y define algunos formatos clave. En la forma más sencilla, el maestro de cada piconet define una serie de ranuras de tiempo de 625 μ seg; las transmisiones del maestro emplean en las ranuras pares y las de los esclavos, en las impares. Esta es la tradicional multiplexión por división de tiempo. Las tramas pueden tener 1,3 o 5 ranuras de longitud. Cada trama se transmite por un canal lógico, llamado enlace, entre el maestro y un esclavo. Hay dos tipos de enlace:

- ACL (Asíncrono no orientado a la conexión): se utiliza para datos commutados en paquetes disponibles a intervalos irregulares. El tráfico ACL se entrega sobre la base de mejor esfuerzo; no hay garantías. Los tramas se pueden perder y tienen que retransmitirse.

- SCO (Síncrono orientado a la conexión): se utiliza para datos en tiempo real. A este tipo de enlace se le asigna una ranura fija en cada dirección. Por la importancia del tiempo en los enlaces SCO, las tramas que se envían a través de ellos nunca se retransmiten.

Administrador de enlace (LMP)

Se encarga de la serialización entre dispositivos Bluetooth para el control de la banda base; establecimiento de la conexión, negociación de parámetros y cambio en las políticas del enlace. Incluye funciones de control de la piconet, la commutación maestro esclavo, el establecimiento de enlaces ACL y SCO y el control de los modos de baja potencia. También se encarga de las funciones de configuración del enlace, como la gestión de la calidad del servicio, el control de potencia y la negociación del tipo de paquetes. Finalmente, tiene también funciones de seguridad, encargándose de la autenticación y el cifrado.

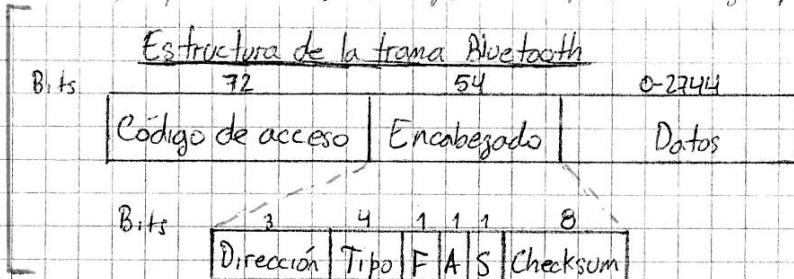
Capa de adaptación y control de enlace lógico (L2CAP)

La capa L2CAP tiene tres funciones principales. Primero, acepta paquetes de hasta 64 kb, provenientes de las capas superiores y los divide en tramas para transmitirlos. Las tramas se reasocian nuevamente en paquetes en el otro extremo.

Segundo, maneja la multiplexión y demultiplexión de múltiples fuentes de paquetes. Cuando

se reensambla un paquete, la capa L2CAP determina cuál protocolo de las capas superiores lo manejará.

Por último, la capa L2CAP se encarga de la calidad de los requerimientos de servicio, fijando el establecimiento de los enlaces como durante la operación normal. Asimismo, durante el establecimiento de los enlaces se negocia el tamaño máximo de carga útil permitido, para evitar saturación. Esto es importante ya que no todos los dispositivos pueden manejar paquetes de 64 kb.



- Código de acceso: identifica al maestro de cada piconet. Su propósito es que los esclavos que se encuentren en el rango de alcance de dos maestros sepan que tráfico está destinado a ellos.

- Encabezado: consta de 54 bits y contiene campos comunes de la subcapa MAC.

• Dirección: identifica a cual de los 8 dispositivos activos está destinada la trama.

• Tipo: indica el tipo de trama (ACL, SCO, sondeo, nula), el tipo de corrección de errores que se utiliza en el campo de datos y cuantas ranuras de longitud tiene la trama.

• F(flow): un esclavo establece este bit cuando su buffer está lleno y no puede recibir más datos. Es una forma primitiva de control de flujo.

• Acknowledgement: se utiliza para incorporar un ACK en la trama.

• S(sequence): sirve para numerar las tramas con el propósito de detectar retransmisiones.

• Checksum: consta de 8 bits y cumple la misma función que en el 802.11.

Todo el encabezado de 18 bits se repite 3 veces. En el receptor se comprobarán las tres copias de cada bit y si coinciden, se acepta. De lo contrario se impone la mayoría.

- Datos: varía desde 240 hasta 2744 bits para una transmisión de 5 ranuras. Para una sola ranura, el campo de datos es de 240 bits.

IP, 4

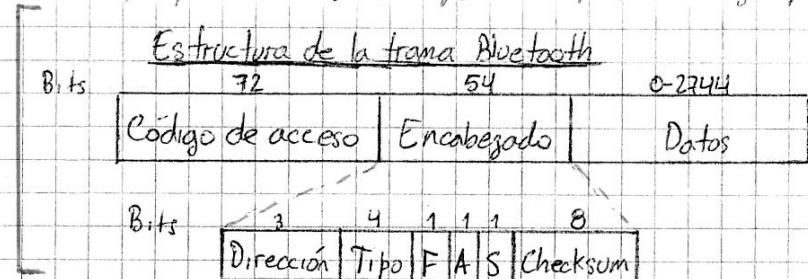
El objetivo del IP es proporcionar una red virtual que comprenda muchas redes físicas interconectadas por routers, así como ofrecer un servicio sin conexión de entrega de paquetes. De esta manera se logra una comunicación transparente sobre diferentes tecnologías.

3 IPv4

Fragmentacion
Direcciones IPv4
Direcciones IPv4
ICMP

se reensambla un paquete, la capa L2CAP determina cuál protocolo de las capas superiores lo manejará.

Por último, la capa L2CAP se encarga de la calidad de los requerimientos de servicio, fijando el establecimiento de los enlaces como durante la operación normal. Asimismo, durante el establecimiento de los enlaces se negocia el tamaño máximo de carga útil permitido, para evitar saturación. Esto es importante ya que no todos los dispositivos pueden manejar paquetes de 64 kb.



- Código de acceso: identifica al maestro de cada piconet. Su propósito es que los esclavos que se encuentren en el rango de alcance de dos maestros sepan que tráfico está destinado a ellos.

- Encabezado: consta de 54 bits y contiene campos comunes de la subcapa MAC.

• Dirección: identifica a cual de los 8 dispositivos activos está destinada la trama.

• Tipo: indica el tipo de trama (ACL, SCO, sondeo, nula), el tipo de corrección de errores que se utiliza en el campo de datos y cuantas ranuras de longitud tiene la trama.

• F(flow): un esclavo establece este bit cuando su buffer está lleno y no puede recibir más datos. Es una forma primitiva de control de flujo.

• Acknowledgement: se utiliza para incorporar un ACK en la trama.

• S(sequence): sirve para numerar las tramas con el propósito de detectar retransmisiones.

• Checksum: consta de 8 bits y cumple la misma función que en el 802.11.

Todo el encabezado de 18 bits se repite 3 veces. En el receptor se comprobarán las tres copias de cada bit y si coinciden, se acepta. De lo contrario se impone la mayoría.

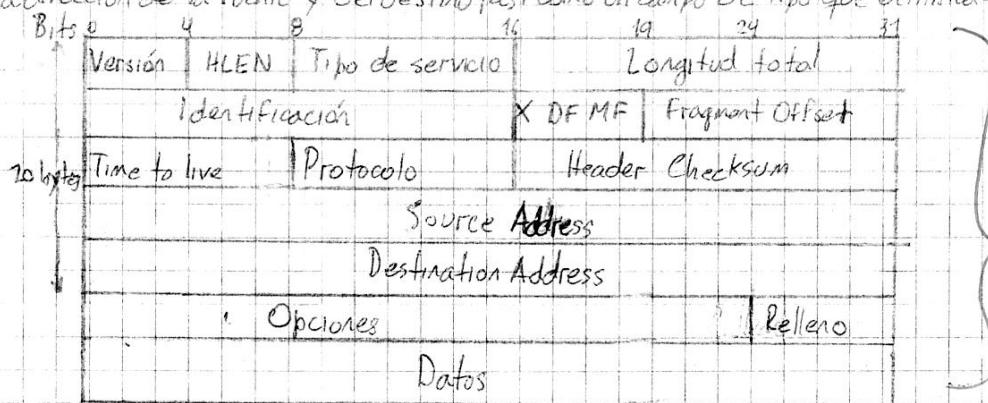
- Datos: varía desde 240 hasta 2744 bits para una transmisión de 5 ranuras. Para una sola ranura, el campo de datos es de 240 bits.

IP, 4

El objetivo del IP es proporcionar una red virtual que comprenda muchas redes físicas interconectadas por routers, así como ofrecer un servicio sin conexión de entrega de paquetes. De esta manera se logra una comunicación transparente sobre diferentes tecnologías.

Datagrama IP

Un datagrama se divide en áreas de encabezado y datos. El encabezado del datagrama contiene la dirección de la fuente y del destino, así como un campo de tipo que identifica su contenido.



- Versión: lleva el registro de la versión del protocolo al que pertenece el datagrama.

- IHL: este campo proporciona la longitud de la cabecera en palabras de 32 bits. Esta es normalmente de 20 bytes, pudiendo llegar a 60 si se utilizan los campos opcionales.

- Tipo de servicio (TOS): este campo especifica cómo debe manejarse el datagrama.

Bits	3	2	1	1	1	2
Prioridad	D	T	R	Sin uso		

• Prioridad: sus valores abarcan desde 0 (prioridad normal) hasta 7 (control de red), permitiendo al emisor indicar la importancia del datagrama. Aún cuando la mayor parte del software de los switches y ruteadores ignora el tipo de servicio, éste es un concepto importante dado que proporciona un mecanismo que permite controlar la información que tendrá prioridad en los destinos.

• D: solicita procesamiento con retardos cortos.

• T: solicita alto desempeño.

• R: solicita alta confiabilidad.

Por supuesto, no es posible para una red garantizar siempre el tipo de transporte solicitado. Por tal motivo debemos pensar en una solicitud de transporte como una indicación para los algoritmos de ruteo, no un requerimiento obligatorio, que el ruteador puede utilizar para seleccionar una ruta con las características más cercanas.

- Longitud total: proporciona la longitud del datagrama medida en octetos. El máximo valor es 65535 bytes, dado que el campo tiene una longitud de 16 bits.

- Identificación: este campo contiene un entero único que identifica al datagrama, y es necesario para que el host destino determine a qué datagrama pertenece un fragmento. Todos los fragmentos de un datagrama contienen el mismo valor de identificación.
- DF: cuando está activado indica que el datagrama no debe fragmentarse.
- MF: este bit especifica si el fragmento contiene datos intermedios del datagrama original, o la parte final.
- Fragment offset: especifica el desplazamiento en el datagrama original de los datos que se están ocurriendo en el fragmento, medido en unidades de 8 bytes, comenzando en 0.
- Time to live: especifica la duración, en segundos, que el datagrama tiene permitido permanecer en el sistema de Internet. Los routers y anfitriones que procesan los datagramas deben decrementar el campo y eliminarlo cuando su tiempo ha concluido. Cada router se configura para decrementar por 1 el campo TTL cuando se procesa el encabezado. Además, se registra el tiempo local cuando llega el datagrama y se decremente el TTL por el número de segundos que el datagrama permanece en el router esperando que se lo despache.
- Protocolo: especifica qué protocolo de alto nivel se utilizó para crear el mensaje que se está transportando en los datos del datagrama.
- Header Checksum: asegura la integridad de los valores del encabezado. Se realiza una suma en bloques de 16 bits de la cabecera y se transmite el complemento a uno como checksum, de forma de tener en el receptor la suma más el checksum igual a 0. Este chequeo se realiza sobre los valores del encabezado y no sobre los datos (sería muy costoso).
- Source Address: contiene la dirección IP del emisor.
- Destination Address: contiene la dirección IP del receptor.
- Opciones: este campo debe ser implementado por todos los dispositivos IP. Generalmente se usan para pruebas de red o depuración.
- Rellenos: representa un grupo de bits puestos en 0 que podrían ser necesarios para asegurar que la extensión del encabezado sea un múltiplo de 32 bits.

Protocolo

Algunos de los posibles valores en el campo protocolo son: ICMP(1), IGMP(2), GGP(3), IP(4), ST(5), TCP(6), EGP(8), UDP(17), ISO-TDP(29), CLNP(80), ISGRP(88), OSPF(89), etc.

Opciones

La longitud de este campo varía dependiendo de qué opción sea seleccionada. Cada opción consiste en un solo octeto de código de opción y a continuación un conjunto de octetos de datos para cada opción.

0	1	3	7
Copy	Option Class	Option Number	

- Copy: controla la forma en que los routers tratan las opciones durante la fragmentación. Cuando está activado, especifica que la opción se debe copiar en todos los fragmentos.

- Option class: especifica la clase general de opción. Puede ser control de red o datagrama (d) o duración y medición (2).

- Option number: establece una opción específica para una clase. Para la clase de control se tiene fin de lista (0, usado si las opciones no terminan al final del header), no operación (1, usado para alinear octetos), seguridad y restricciones de manejo (2, para aplicaciones militares), routers no estricto de fuente (3, usado para rutear un datagrama a través de una trayectoria específica), registro de ruta (7), identificador de flujo (8, obsoleto), y router estricto de fuente (9). Para la clase de duración se tiene sello de tiempo Internet (4, usado para registrar sellos de hora en una ruta).

Opción de registro de ruta

Esta opción permite a la fuente crear una lista de direcciones IP y arreglarla para que cuando un router que maneje el datagrama agregue su propia dirección a la lista.

0	8	16	31
Código	Longitud	Puntero	
		Primera dirección IP	
		Segunda dirección IP	

- Código: explicado anteriormente. En este caso contiene 0 en la clase de opción y 7 en su número.

- Longitud: especifica la longitud total de la opción.

- Puntero: especifica el desplazamiento, dentro de la opción, de la siguiente ranura disponible.

- Primera dirección IP: campo reservado para comenzar a registrar direcciones.

Cada vez que una máquina maneja un datagrama que tiene la opción de registro de ruta activada, añade su dirección a la lista (se debe reservar suficiente espacio desde la fuente original para manejar tantas las enfraldas necesarias). La máquina combina el puntero y el can-

ps de longitud; si el puntero es mayor significa que la lista está llena y se pasa el datagrama sin incluirse. En caso contrario, la máquina inserta su dirección en la posición especificada por el puntero y luego incrementa en 4 el valor de éste.

Cuando un datagrama llega a su destino, la máquina puede extraer y procesar la lista de direcciones IP. Para usar la opción de registro de ruta se requiere de dos máquinas que estén de acuerdo para cooperar, ya que la opción no se activa de manera automática.

Opción de ruta fuente

Esta opción proporciona al enrutador una forma para determinar una ruta a través de la Internet (por ejemplo, para probar desempeños). El formato de la opción es similar a la anterior, pero en este caso la lista de direcciones contiene los saltos a realizar.

IP soporta dos formas de ruteo de fuente. Una forma, conocida como ruteo estricto de fuente, establece una vía de ruteo que debe seguirse exactamente. La ruta entre dos direcciones sucesivas de la lista debe consistir en una sola red física; se producirá un error si el router no puede seguir una ruta estricta de fuente.

La otra forma, conocida como ruteo no estricto de fuente, permite múltiples saltos de redes entre direcciones sucesivas de la lista.

Ambas opciones requieren que los routers a lo largo de la trayectoria anoten su propia dirección de red local en la lista de direcciones. De esta forma se obtiene una lista con todas las direcciones recorridas, similar a la opción de registro de ruta.

Cada router debe comprobar si la lista está completa; en tal caso establece la ruta del datagrama hacia su destino como lo hace normalmente. De otra manera, reemplaza la dirección asignada en el puntero por la suya (la que corresponde a la red actual) y establece la dirección para el datagrama usando la dirección que obtuvo de la lista.

Opción de sello de hora

Esta opción trabaja como la de registro de ruta. Contiene una lista inicial vacía y cada router a lo largo de la ruta escribe sus datos en la lista.

Cada entrada de la lista contiene dos datos de 32 bits: la dirección IP del router que proporciona la entrada y un entero de sello de hora de 32 bits, que define la hora y fecha en la que el router maneja el datagrama, expresado en milisegundos desde la medianoche. Tiempo

Universal.

Código	Longitud	Puntero	Oflow	Flags	
Primera dirección IP					
Primer sello de hora					

- Código: en este caso su valor es 68.

- Longitud: especifica la longitud del espacio reservado para la opción.

- Puntero: indica la posición de la siguiente ranura no utilizada.

- Oflow: contiene un contador enero de routers que podrían no registrar sellos de hora por falta de espacio.

- Flags: controla el formato exacto de la opción y establece cómo los routers deben suministrar el sello de hora. En 0 se registran sólo los sellos de hora, en 1 se anteponen las direcciones IP y en 3 se da una lista de direcciones IP y sólo se añade el sello de hora si la próxima dirección en la lista coincide con la próxima dirección del router.

Fragmentación

En un caso ideal, el datagrama IP completo se ajusta dentro de una trama física haciendo que la transmisión a través de la red física sea eficiente. Para alcanzar esta eficiencia, los diseñadores tendrían que seleccionar un tamaño máximo de datagrama para que el datagrama siempre se ajuste dentro de la trama.

Cada tecnología de conmutación de paquetes establece un límite superior fijo para la cantidad de datos que pueden transferirse en una trama física. Nos referimos a estos límites como la cantidad de transferencia máxima (MTU) de una red (por ejemplo, Ethernet limita a 1500 octetos). Toda red debe aceptar un MTU de al menos 68 bytes (para tener 8 de datos), aunque lo recomendado es 576 bytes.

La limitación de los datagramas para que se ajusten a la MTU más pequeña posible en una red de redes hace que la transferencia sea ineficiente cuando estos datagramas pasan a través de una red que puede transportar tramas de mayor tamaño.

En lugar de diseñar datagramas que se ajusten a las restricciones de la red física, el software TCP/IP selecciona una tamaña de datagrama más conveniente desde el principio y establecer un for-

ma para dividir datagramas en pequeños fragmentos cuando el datagrama necesita viajar a través de una red que tiene una MTU pequeña. Las pequeñas piezas dentro de un datagrama dividido se conocen como fragmentos y el proceso de división se conoce como fragmentación.

Por lo general, la fragmentación se da en un router a lo largo del trayecto entre la fuente del datagrama y su destino final, cuando hay diferencias de MTU entre las redes que maneja el router.

El tamaño de cada fragmento se selecciona de manera que cada uno de estos pueda transportarse a través de la red subyacente en una sola trama. Dado que IP representa el despliegamiento de datos en múltiplos de 8 octetos, el tamaño del fragmento debe ser múltiplo de 8. Por supuesto, al elegir el múltiplo más cercano al MTU de la red, no es usual dividir el datagrama en fragmentos de igual tamaño; los últimos fragmentos son generalmente más cortos que los otros. Los fragmentos deben reensamblarse para producir una copia completa del datagrama original, antes de que pueda procesarse en su lugar de destino.

El protocolo IP no limita los datagramas a un tamaño pequeño, ni garantiza que los datagramas más grandes serán entregados sin fragmentación. La fuente puede seleccionar cualquier tamaño de datagrama que considere apropiado; la fragmentación y el reensamblado se dan automáticamente sin que la fuente deba realizar ninguna acción especial.

Cada fragmento contiene un encabezado de datagrama que duplica la mayor parte del encabezado del datagrama original, seguido por tantos datos como pueden ser acarreados en el fragmento) mientras la longitud total se mantiene en un valor menor al MTU de la red en la que debe viajar.

En una red de redes TCP/IP, una vez que un datagrama se ha fragmentado, los fragmentos viajan como datagramas separados hacia su destino final donde serán reensamblados. Esto tiene dos desventajas. Primero, si hubo un tramo de la red con MTU pequeña, los pequeños fragmentos deben transportarse en esa forma desde el punto de fragmentación hasta el destino final. Segundo, si se pierde cualquier fragmento, el datagrama no podrá reensamblarse. La máquina de recepción activa un temporizador de reensamblado al recibir un fragmento inicial. Si el temporizador termina antes de que lleguen todos los fragmentos, se descartan los fragmentos sin procesar el datagrama. Así, la probabilidad de perder un datagrama se incrementa con la fragmentación.

El control de la fragmentación se realiza con los campos de la cabecera Identificación, DF, MF y Fragment Offset.

```

Pseudocódigo: IP_Fragmentation_Module(datagram)
    extract size of datagram
    IF (size > MTU of corresponding network)
        IF (D bit is set)
            discard datagram
            send ICMP error message
            return
        }
        else
            calculate maximum size
            divide segment into fragments
            add header to each fragment
            add required options to each fragment
            send fragments
            return
        }
    }
    else send the datagram
    return
}

```

Direcciones IP

Cada anfitrión en la Internet tiene asignada una dirección de número entero de 32 bits, llamada su dirección IP. La parte inteligente del direccionamiento en una red de redes es que los números enteros son seleccionados con cuidado para hacer eficiente el ruteo. Una dirección IP codifica la identificación de la red a la que se conecta el anfitrión, así como la identificación de un anfitrión único en esa red.

Los 32 bits de una dirección IP se separan en 4 grupos de 8 bits y así se obtiene una representación decimal de cada dirección, simplificando su lectura y escritura.

Conceptualmente, cada dirección es un par (net id, host id), en donde net id identifica una red y host id un anfitrión dentro de la red. Existen 5 clases de direcciones IP:

- Clase A: su primer bit es un 0. Asignan 7 bits al net id, y 24 al host id.
- Clase B: sus primeros dos bits son 1 y 0. Asignan 16 bits al net id y 16 al host id.
- Clase C: sus primeros tres bits son 1,1 y 0. Asignan 21 bits al net id y 8 al host id.
- Clase D: sus primeros cuatro bits son 1,1,1 y 0. Se utilizan para multicast.
- Clase E: sus primeros cinco bits son 1,1,1,1 y 0. Están reservadas para uso futura.

Dado que las direcciones IP codifican tanto una red y un anfitrión en dicha red, no especifican una computadora individual, sino una conexión a la red.

Las direcciones de Internet pueden utilizarse para referirse a redes así como a anfitriones individuales. Por regla, una dirección que tiene todos los bits del campo host id en 0, se reserva para referirse a la red en sí misma.

Las direcciones IP se pueden utilizar para especificar la difusión; estas direcciones se transforman en difusión por hardware, si ésta se encuentra disponible. Por regla, una dirección de difusión tiene todos los bits del campo ^{host id} en 1.

Técnicamente, la dirección de difusión que describimos se conoce como dirección de difusión directa, debido a que contiene tanto una identificación válida de red como el campo host id de difusión. Este tipo de direcciones se puede interpretar sin ambigüedades en cualquier punto de la Internet ya que identifica en forma única la red objetivo, además de especificar la difusión en dicha red. Como desventaja, se debe conocer la dirección de la red.

Otra forma de dirección de difusión, llamada dirección de difusión limitada, proporciona una dirección de difusión para la red local, independientemente de la dirección IP asignada. La dirección de difusión local consiste en todos los bits puestos en 1.

Una dirección IP con campo host id 0 se refiere a "este" anfitrión, y una dirección de Internet con el net id 0 se refiere a "esta" red. Pero sólo es significativo utilizar direcciones de esta forma cuando no hay ambigüedad.

Máscaras

Las máscaras se inventaron cuando se decidió dejar de usar direcciones con clases para pasar a la forma más flexible de las subredes y supernetes. Las máscaras sirven para identificar el netid de una dirección y generalmente consisten en una secuencia de 1s. Se utiliza una notación de punto que indica cuántos 1 tiene la submáscara. Por ejemplo, la máscara de subred de una dirección clase A es /8, de una clase B /16 y de una clase C /24.

Al efectuar un AND lógico entre una dirección IP y su máscara, se obtiene el campo net id de la dirección.

Direcciones IP especiales

No todas las direcciones comprendidas entre 0.0.0.0 y 223.255.255.255 son válidas para un host; algunas de ellas tienen significados especiales.

- 0.0.0.0: se envía en el arranque para obtener una dirección IP automática mediante DHCP.

- 0 en net id, host: indica un host en la red actual.
- net. 0 en host: identifica una red.
- 255.255.255.255: difusión limitada.
- net. 255 en host: difusión a la red indicada.
- 127.x.x.x: loopback. Se utiliza para comprobar que los protocolos TCP/IP estén instalados correctamente en un ordenador.
- 10.0.0.0/8: direcciones privadas. Este tipo de direcciones sólo son visibles por otros hosts de su propia red o de otras redes privadas interconectadas por routers. PCs con direcciones privadas pueden salir a Internet por medio de un router con IP pública; pero desde Internet no se puede acceder a una IP privada.
- 172.16.00-172.31.00/16: direcciones privadas. $172.16.00/16 \approx 172.31.0.0/16$
- 192.168.0.0/16: direcciones privadas. $192.168.0.0/16 \approx 192.168.2.55.6/24$

Ruteo

En un sistema de conmutación de paquetes, el ruteo es el proceso de selección de un camino sobre el que se mandan paquetes y el rutador es la computadora que hace la selección. El algoritmo de ruteo IP debe escoger cómo enviar un datagrama pasando por muchas redes físicas. La mayor parte del software de ruteo es bastante simple y selecciona rutas basándose en suposiciones sobre los caminos más cortos.

Podemos dividir el ruteo en dos tipos: entrega directa (la transmisión de un datagrama desde una máquina hasta otra a través de una sola red física) y entrega indirecta (cuando el datagrama debe pasar a través de un router para su entrega).

Entrega directa

La transmisión de un datagrama IP entre dos máquinas dentro de una sola red física no involucra routers. El transmisor encapsula el datagrama dentro de una trama física, transforma la dirección IP de destino en una dirección física de hardware (MAC, usando ARP si no la tiene) y envía la trama resultante directamente a su destino.

Para averiguar si un destino reside en una de las redes directamente conectadas, el transmisor extrae la porción de red de la dirección IP de destino y la compara con la porción de red de

Compara el net id de D'Addos y su netid

su propia dirección IP; si corresponden, el datagrama se puede enviar de forma directa. Esta comprobación es muy eficiente debido al esquema de direccionamiento de Internet (prefijos comunes y extracción simple).

Entrega indirecta

Este tipo de entrega es más difícil que la directa porque el transmisor debe identificar un router para enviar el datagrama. Cada vez que la trama llega a un router, el software extrae el datagrama encapsulado y el software IP selecciona el siguiente router a lo largo del camino hacia el destino. El primer y último paso se realizan por entrega directa.

Ruteo IP controlado por tabla

El algoritmo usual de ruteo IP emplea una tabla de ruteo en cada máquina que almacena información sobre posibles destinos y sobre cómo alcanzarlos. De modo a que tanto los routers como los anfitriones ruten datagramas, ambos tienen tablas de ruteo. Para ocultar información y mantener la eficiencia del proceso, las tablas de ruteo sólo necesitan contener prefijos de red y no direcciones completas.

Ruteo con salto al siguiente

Por lo común, una tabla de ruteo contiene pares (N,R), donde N es la dirección IP de una red de destino y R la dirección IP del "siguiente" router en el camino hacia la red N. El router R es conocido como el salto siguiente y la idea general se conoce como ruteo con salto al siguiente. Por lo tanto, la tabla de ruteo en R sólo especifica un paso a lo largo del camino de R a su red de destino; el router no conoce el camino completo hacia el destino.

Es importante el hecho de que cada registro en una tabla de ruteo apunta hacia un router que reside en una red que el router puede alcanzar en forma directa. Cuando un datagrama viaja dejando un router, el software IP localiza la dirección de destino y extrae la porción de red, que el router utiliza para tomar una decisión de ruteo.

Este sistema tiene algunos inconvenientes, como la utilización de un solo anillo que podría no ser óptimo, cierta dificultad para reportar problemas de entrega, o diferentes de anillo para enviar y recibir entre dos máquinas específicas.

Rutas asignadas por omisión

Otra técnica utilizada para ocultar información y mantener reducido el tamaño de las tablas de ruteo es asociar muchos registros a un router asignado por omisión. La idea es hacer que el software de ruteo IP busque primero la tabla de ruteo para encontrar la dirección de destino. Si no apare-

ce una ruta en la tabla, las rutinas de ruteo envían el datagrama a un router asignado por omisión.

Sobredes

Esta técnica se utiliza para permitir que una sola dirección de red abarque muchas redes físicas.

Se piensa que una dirección IP tiene una porción de Internet y una porción local, en donde la porción de red identifica una localidad (posiblemente con muchas redes físicas) y la porción local identifica una red física y su anfitrón en dicha localidad.

El resultado es una forma de direccionamiento jerárquico que lleva al cortes por parte de ruteo jerárquico. Una red con sobredes es vista desde fuera como una sola red.

Para permitir una máxima flexibilidad al partitionar las direcciones de subred, el estándar TCP/IP de subred permite que la interpretación se elija de forma independiente para cada red física. El estándar especifica que una localidad que utiliza el direccionamiento de subred debe escoger una máscara de subred para cada red, para así poder identificarlas con facilidad.

Ruteo con sobredes

Se debe modificar el algoritmo estándar de ruteo IP para trabajar con direcciones de subred. Para lograr un ruteo óptimo, una máquina M debe utilizar el ruteo de subred para una dirección IP de red N, a menos que exista un solo camino que sea el más corto entre M y cualquier red física que sea subred de N. Pero esta restricción teórica no ayuda en la asignación de sobredes (los caminos más cortos pueden cambiarse imposible prohibir rutas de subred fuera de fronteras).

El algoritmo de ruteo en subredes basa sus decisiones en una tabla de ruteo. En las subredes, no es posible decidir qué bits corresponden a la red ni cuáles corresponden al anfitrón sólo con la dirección. El algoritmo modificado que se utiliza con las subredes guarda información adicional en la tabla de ruteo. Cada registro dentro de la tabla contiene un campo adicional que especifica la máscara de subred utilizada.

Cuando el algoritmo elige rutas, utiliza la máscara de subred para extraer bits de la dirección de destino y compararlos con el registro en la tabla, mediante un AND entre la máscara y la dirección de destino del datagrama. Si los valores coinciden, rutea el datagrama hacia la dirección del siguiente salto.

ICMP

En el sistema sin conexión de IP, cada router opera de manera autónoma, reteniendo o entregando los datagramas que llegan, sin coordinarse con el transmisor original. El sistema trabaja bien si todas las máquinas funcionan de manera correcta y si están de acuerdo respecto a las rutas. Pero ningún sistema funciona bien todo el tiempo. Además de las fallas en las líneas de comunicación y en los procesadores, el IP tiene fallas en la entrega de datagramas cuando la máquina de destino está desconectada de la red, cuando el TTL expira, o cuando hay mucha congestión en los routers intermedios. El protocolo IP, por sí mismo, no contiene nada para ayudar al transmisor a comprobar la conectividad ni para ayudarlo a aprender sobre fallas en la entrega.

Para permitir que los routers reporten los errores o proporcionen información sobre circunstancias no esperadas, los diseñadores agregaron a los protocolos TCP/IP un mecanismo de mensajes de protocolo especial, conocido como Protocolo de Mensajes de Control Internet (ICMP).

Al igual que el resto del tráfico, los mensajes ICMP viajan a través de la Internet en la porción de datos de los datagramas IP. Sin embargo, el destino final de un mensaje ICMP no es un programa ejecutado en un usuario en la máquina de destino, sino el software IP en dicha máquina.

El ICMP no se restringe sólo a los routers; cualquier máquina puede enviar un mensaje ICMP a otra, proporcionando un mecanismo que se utiliza para todos los mensajes de información y control.

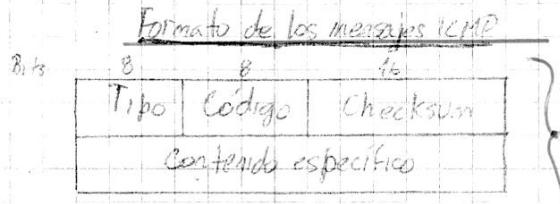
Cuando un datagrama causa un error, el ICMP sólo puede reportar la condición del error a la fuente original del datagrama; la fuente debe relacionar el error con un programa de aplicación individual o debe tomar alguna otra acción para corregir el problema. Se reporta a la fuente original debido a que un datagrama sólo especifica la dirección de ésta y del destino final.

Entrega de mensajes ICMP

Los mensajes ICMP requieren dos niveles de encapsulación, ya que cada mensaje ICMP viaja a través de la Internet en la porción de datos de un datagrama IP, el cual viaja a través de cada red física en la porción de datos de una trama. Los mensajes ICMP también pueden perderse o causar congestionamiento adicional. Cuando esto ocurre, no se genera un mensaje ICMP de error; para evitar el problema de tener mensajes de error sobre mensajes de error.

Hay que tener en cuenta que aunque los mensajes ICMP se encapsulan y envían mediante el IP,

ICMP no se considera como un protocolo de nivel más alto sino como una parte obligatoria del IP.



- Tipo: identifica el mensaje. Puede ser:

- 0/8: respuesta/solicitud de eco.
- 3: destino inaccesible.
- 4: disminución de origen.
- 5: redirección (cambiar una ruta).
- 9/10: descubrimiento/aviso de router.
- 11: tiempo excedido.
- 12: problema de parámetros.
- 13/14: solicitud/respuesta de timestamp.
- 17/18: solicitud/respuesta de máscara.

- Código: proporciona más información sobre el tipo de mensaje.

- Checksum: cubre sólo el mensaje ICMP. Usa el mismo algoritmo que IP.

Además, los mensajes ICMP que reflejan errores siempre incluyen el encabezado y los primeros 64 bits de datos del datagrama que causó el problema, para permitir que el receptor determine de manera más precisa los protocolos y programas de aplicación responsables del datagrama.

Prueba de accesibilidad y estado de un destino (Ping)

Una de las herramientas de depuración más utilizadas incluye los mensajes ICMP de echo request y echo reply. Un host o un router envía un mensaje ICMP de echo request hacia un destino específico. Cualquier máquina que recibe una solicitud de eco, formela una respuesta y la regresa al transmisor original. La solicitud contiene un área opcional de datos, copiada en la respuesta. La solicitud de eco y su respuesta asociada se pueden utilizar para comprobar si un destino es alcanzable y si responde. La recepción exitosa de una respuesta verifica que los piezas principales del sistema de transporte están funcionando bien.

En muchos sistemas el comando que llama el usuario para enviar solicitudes de eco ICMP se conoce como ping. Los versiones más sofisticadas de ping envían una serie de solicitudes ICMP, capturan las respuestas y proporcionan estadísticas sobre la pérdida de datagramas.

Bits 0	7	15	31
Tipo: 0/8	Código: 0	Checksum	
Identificador	Número de secuencia		
Datos opcionales			

- Identificador: usado por el transmisor para responder las solicitudes.

- Número de secuencia: idem.

- Datos opcionales: campo de longitud variable. Contiene los datos que se regresan al transmisor.

Reporte de destinos no accesibles

Cuando un router no puede direccionar o entregar un datagrama IP, envía un mensaje de destino inaccesible a la fuente original.

Bits 0	8	8	16
Tipo: 3	Código: 0-12	Checksum	
No utilizado			
IP header + 64 data bits			

El campo Código contiene un número entero que describe con más detalle el problema. Puede ser: red inaccesible (0), host inaccesible (1), protocolo inaccesible (2), puerto inaccesible (3), fragmentación necesaria pero DF activado (4), fallo en la ruta de origen (5), red de destino desconocida (6), anfitrión de destino desconocido (7), anfitrión de origen aislado (8), comunicación con la red/anfitrión de destino administrativamente prohibida (9/10) o red/anfitrión inaccesible por el tipo de servicio (11/12).

Aunque IP es un mecanismo de entrega con el menor esfuerzo, el descarte de datagramas no debe tomarse a la ligera. Siempre que un error evite que un router direccione o entregue un datagrama, el router envía al origen un mensaje de destino no accesible y luego suelta el datagrama. Por lo general estos errores implican fallas en el ruteo. Como se incluye un prefijo del datagrama erróneo, la fuente sabrá que dirección no es accesible.

Aunque los routers reportan las fallas que encuentran, quizás no tengan conocimiento de todas las fallas de entrega.

Control de congestionamiento y de flujo

Dado a que IP funciona sin conexión, un router no puede reservar memoria o recursos de comunicación antes de recibir datagramas. Como resultado, los routers se pueden saturar con

el tráfico y condición conocida como congestión. El congestionamiento puede surgir por las razones diferentes: una cambiar de alta velocidad capaz de generar tráfico más rápido de lo que una red lo puede transferir o muchas comunicadoras que necesitan enviar datagramas al mismo tiempo a través de un solo router.

Cuando los datagramas llegan demasiado rápido para que su enrutador o un router los procesen, estos los tienen temporalmente en una cola de espera en memoria. Este procedimiento de memorización temporal soluciona el problema para paquetes pequeños de datagramas. Si el tráfico continúa llega en momento en el que se le acaba la memoria al enrutador o al router, y deben descartar los demás datagramas que lleguen. Una máquina utiliza mensajes ICMP de source quench para reportar el congestionamiento a la fuente original. Este mensaje es una solicitud para que la fuente reduzca la velocidad de transmisión de datagramas. Por lo general, los routers congestionados envían un mensaje de source quench por cada datagrama que descartan. Algunos, más sofisticados, reducen las fuentes con velocidades más altas o emplean a enviar source quenches cuando sus colas de espera crecen, pero antes de que se saturen.

No existe ningún mensaje ICMP para revertir el efecto de un source quench. En cambio, en enrutador que reciba mensajes de disminución para un destino D, baja la velocidad de envío de datagramas hacia D, hasta que deja de recibir mensajes de source quench; luego, aumenta la velocidad de manera gradual en tanto no reciba más solicitudes de source quench.

El formato de un mensaje de source quench es el mismo que el de un mensaje de destino no accesible: incluye un prefijo del datagrama que activó la solicitud.

Solicitud para cambio de ruta

Por lo general las tablas de ruteo se mantienen sin cambios por grandes períodos de tiempo. Los enrutadores las initian desde un archivo de configuración en el arranque del sistema y los administradores de sistemas hacen cambios de ruteo durante la operación normal esperadamente. Si cambia la topología de la red las tablas de ruteo pueden volverse incorrectas. Los routers intercambian información de ruteo en forma periódica para incorporar los cambios en la red y mantener actualizadas sus rutas. Como regla general se assume que los routers conocen rutas correctas; los hosts

convergen con información mínima de ruteo y aprender nuevas rutas de los routers.

En un caso especial, cuando un router detecta un anfitrión que utiliza una ruta no óptima, le envía al anfitrión un mensaje ICMP Redirect, solicitándole que cambie sus rutas. La ventaja del esquema de redireccionamiento ICMP es la simplicidad: permite que un anfitrión intere considerando solamente un router en la red local. El router inicial genera mensajes redirect siempre que un anfitrión envíe un datagrama para el que existe una ruta mejor. La tabla de ruteo del host permanece reducida y así así contiene rutas óptimas para todos los destinos en uso.

Bits	8	8	16
Tipo:	5	Código: 0-3	Checksum
Router IP Address			
IP header + 64 data bits			

- Código: especifica con mayor detalle cómo interpretar la dirección de destino. Puede referirse a redireccionamiento de datagramas para la red(0), el anfitrión(1), el tipo de servicio y la red(2) o el tipo de servicio y el anfitrión(3).

- Router IP Address: contiene la dirección de un router que el anfitrión utilizará para alcanzar el destino deseado.

Detección de rutas circulares o buzos

Debido a que los routers en Internet computan un salto al siguiente router, utilizando tablas locales, los errores en dichas tablas pueden producir un ciclo de ruteo. Si un datagrama entra en un ciclo de ruteo, recorrerá indefinidamente y de manera circular todos los routers del ciclo. Recordemos que cada datagrama IP tiene un contador de tiempo de vida que disminuye al ser procesado por un router y al llegar a 0 ocurre el descarte del datagrama.

Siempre que un router descarta un datagrama porque su TTL llega a 0 o porque el contador de reensamblado llega a 0 envía un mensaje ICMP de tiempo excedido a la fuente del datagrama.

El formato del mensaje es el mismo que el de un mensaje de destino no accesible. El código es 0 cuando el TTL llega a 0 y 1 cuando el tiempo para el reensamblado de fragmentos expiró.

Reporte de otros problemas

Cuando un router o un host encuentran problemas que no se han cubierto con los mensajes ICMP de error anteriores, envían un mensaje de problema de parámetros a la fuente original (una cursiva puede ser argumentos para una opción incorrecta). El mensaje sólo se envía cuando el pro-

Problema es tan severo que se tiene que descartar el datagrama.

bits	8	9	16
Tipo:	12	Código 0/1	Checksum
Pointer		No utilizado	
D	Habilit + 8 bits de datos		

- Código: 1 para informar que falta la ejecución requerida (ej: seguridad en red militar).

- Pointer: identifica el octeto del datagrama que causó el problema. No se usa con código 1.

Marcas de tiempo

Aunque las máquinas de Internet se pueden comunicar, por lo general operan de forma independiente, con cada máquina manteniendo su propia noción de hora local. Esto puede traer problemas en sistemas distribuidos. Una de las técnicas más sencillas para sincronizar los relojes se vale de un mensaje NTP para obtener la hora de otra máquina, que se llamará solicitud de timestamp. La máquina receptora enviará respuesta de timestamp a quien la solicitó.

bits	0	7	15	31
Tipo:	13/14	Código 0	Checksum	
Identificador		Número de secuencia		
Original timestamp				
Recibir timestamp				
Transmitir timestamp				

- Tipo: 13 (solicitud) o 14 (respuesta).

- Identificador: utilizado por la fuente para asociar las solicitudes con las respuestas.

- Número de secuencia: idem.

- Original timestamp: llenado por la fuente original justo antes de transmitir el paquet.

- Recibir timestamp: llenado inmediatamente al recibir una solicitud.

- Transmitir timestamp: llenado justo antes de transmitir la respuesta.

Los antifirmeos usan estos 3 campos para computar estimaciones del tiempo de retraso entre ellos y para sincronizar sus relojes. Para esto computa el tiempo total de viaje de la solicitud, el tiempo que estuvo en la máquina destino y el tiempo de tránsito en la red y estimar las diferencias entre los relojes.

Obtención de una máscara de subnet

Para participar en el reclamamiento de subnet, un host necesita saber qué bits de la dirección IP corresponden a la red física así como qué bits corresponden al host id. La información necesaria

para interpretar la dirección se representa en una máscara de subred.

Para aprender la máscara de subred utilizada para la red local; una máquina puede enviar un mensaje de solicitud de máscara de subred a un router y recibir una respuesta de máscara de subred. La máquina que hace la solicitud puede enviar directamente el mensaje si conoce la dirección del router, o transmitir el mensaje por difusión.

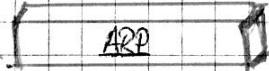
Datos	?	15	31
Tipo: 17/18	Código: 0	Checksum	
Identificador	Número de secuencia		
Máscara de dirección			

- Tipo: 17 (solicitud) o 18 (respuesta).

- Identificador: usado para asociar las solicitudes con las respuestas.

- Número de secuencia: idem.

- Máscara de dirección: contiene la máscara de subred en la respuesta.



Consideremos dos máquinas, A y B, que comparten una red física. Cada una tiene asignada una dirección IP, I_A e I_B , así como una dirección física, P_A y P_B . El objetivo es diseñar en software de bajo nivel que encalle las direcciones físicas y permita que programas de un nivel más alto trabajen sólo con direcciones de Internet. Sin embargo, la comunicación debe llevarse a cabo por medio de redes físicas, utilizando el esquema de direcciones físicas proporcionado por el hardware. Si A quiere enviar un paquete a B, pero sólo tiene su dirección IP I_B , ¿cómo puede obtener la dirección P_B ?

La transformación de direcciones se tiene que realizar en cada fase a lo largo del camino. En particular, surgen dos casos. Primero, en la última fase de entrega de un paquete, éste se debe enviar a través de una red física hacia su destino final; o sea que la computadora que envía el paquete tiene que transformar la dirección IP de destino final en su dirección física. Segundo, en cualquier punto del camino que no sea la fase final, el paquete se debe enviar hacia un router intermedio, por lo que el transmisor tiene que transformar la dirección IP del router en una dirección física.

El problema de transformar direcciones de alto nivel en direcciones físicas se conoce como problema de asociación de direcciones. Los diseñadores de los protocolos TCP/IP encontraron una solución creativa para redes como Ethernet, que tiene capacidad de difusión. Para evitar la definición de una tabla de conversiones, los diseñadores utilizan un protocolo de bajo nivel para asignar direcciones en forma dinámica; conocido como Protocolo de Asociación de Direcciones (ARP), que pro-

proporciona un mecanismo eficaz y fácil de mantener.

La idea es muy sencilla: cuando la máquina A quiere resolver la dirección IP de B, transmite por difusión un paquete especial, que pide al host que tiene la dirección IP que responda con su dirección física. Todos los hosts reciben la solicitud, pero sólo el host B reconoce su propia dirección IP y envía una respuesta que contiene su dirección física. Cuando A recibe la respuesta, utiliza la dirección física para enviar el paquete IP a B.

En resumen, ARP permite que un host encuentre la dirección física de otro host dentro de la misma red física con sólo proporcionar la dirección IP de su objetivo.

Memoria intermedia

Puede parecer extraño que para que A envíe un paquete a B, primero tenga que transmitir una difusión que llegue a B. Podría pensarse que es mejor enviar por difusión el paquete que quiere entregar, pero la difusión es demasiado cara para usarse cada vez que una máquina necesita transmitir un paquete a otra, ya que requiere que cada máquina en la red procese dicho paquete.

Para reducir los costos de comunicación, las computadoras que utilizan ARP mantienen una memoria intermedia de las asignaciones de direcciones IP a dirección física recientemente adquiridas, para evitar usar ARP varias veces. Siempre que una computadora recibe una respuesta ARP, ésta guarda la dirección IP del transmisor, así como la dirección de hardware correspondiente, en su memoria intermedia. Cuando transmite un paquete, una computadora siempre busca una asignación en su memoria inmediatamente de enviar una solicitud ARP.

Refinamientos

Un hecho a tener en cuenta es que si el host A va a dirigir A a B porque necesita enviar algo a B, existe una alta posibilidad de que B necesite enviar algo a A en un futuro cercano. Para anticifarse la necesidad de B y evitar tráfico de red adicional, A incluye su asignación de dirección IP como dirección física cuando envía una solicitud a B. Éste extrae la asignación de A de la solicitud, la graba en su memoria intermedia ARP y envía la respuesta hacia A.

Debido a que A transmite por difusión su solicitud inicial, todas las máquinas en la red la oyeron y pueden extraer, así como guardar en su memoria intermedia, la asignación de A.

Cuando una máquina cambia su dirección física, las otras computadoras en la red, que tienen almacenada una asignación en su memoria intermedia ARP, necesitan ser informadas para que puedan cambiar el registro. Un sistema puede notificar a otros sobre una nueva dirección física al enviar una difusión ARP cuando se inicia.

Hay que tener en cuenta que ARP se piensa como parte del sistema físico de red, no como parte de los protocolos de Internet.

Implementación de ARP

De manera funcional, ARP está dividido en 2 partes. La primera parte transforma una dirección IP en una dirección física cuando se envía un paquete, y la segunda responde solicitudes de otras máquinas. Al tener una dirección IP de destino, el software consulta su memoria intermedia ARP para encontrar la transformación de la dirección. Si la conoce, el software extrae la dirección física, pone los datos en una trama utilizando esa dirección y envía la trama. Si no conoce la transformación, el software debe transmitir una solicitud ARP por difusión y esperar la respuesta.

La difusión de una solicitud ARP se puede volver compleja. La máquina de destino puede estar apagada o muy ocupada para aceptar la solicitud. También se puede perder la solicitud ARP inicial y en tal caso deberá retransmitirse. Mientras tanto el anfitrión tiene que almacenar el paquete original, pero si el retraso es significativo, el anfitrión puede descartar los paquetes salientes. Es importante tener software ARP que maneje de manera temporal la tabla de asignaciones y que remueva los registros después de un período establecido de tiempo.

La segunda parte del código ARP maneja paquetes que llegan por medio de la red. Cuando llega un paquete ARP, el software extrae la dirección IP del transmisor y la dirección física, luego verifica la memoria temporal para verificar si ya existe un registro para el transmisor. Si es así, el controlador actualiza el registro al sobreescibir la dirección física con la dirección obtenida del paquete. Después, el receptor procesa el resto del paquete ARP.

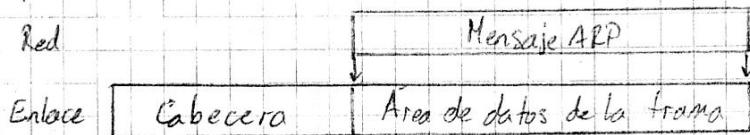
El receptor debe manejar dos tipos de paquetes ARP entrantes. Si llega una solicitud ARP, la máquina receptora debe verificar si es el objetivo de la solicitud. Si es así, el software ARP formula una respuesta al proporcionar su dirección física y la envía directamente al solicitante.

El otro caso interesante sucede cuando llega una respuesta ARP. El controlador crea un registro en la memoria temporal y luego el receptor intenta encontrar una correspondencia entre la

respuesta y la solicitud enviada anteriormente. Si en el tiempo entre que una máquina transmite una solicitud ARP y recibe la respuesta, se generan solicitudes adicionales para la misma dirección, el software debe recordar que ya envió una solicitud para no enviar más.

Encapsulación de ARP

Cuando los mensajes ARP viajan de una máquina a otra, se deben transportar en tramas físicas



Para identificar que la trama transporta un mensaje ARP, el transmisor asigna un valor específico al campo de tipo en la cabecera de la trama (ej.: Ethernet 0806) y coloca el mensaje ARP en el campo de datos de la trama. Cuando la trama llega a una computadora, el software de red utiliza el campo de tipo de trama para determinar su contenido.

Formato de ARP

Los datos en los paquetes ARP no tienen un encabezado con formato fijo. Por el contrario, para hacer que ARP sea útil para varias tecnologías de red, la longitud de los campos con direcciones depende del tipo de red. Pero, para hacer posible la interpretación de un mensaje ARP arbitrario, se incluyen campos fijos cerca del comienzo, que especifican la longitud de las direcciones.

Bits	0	7	15	31
	Tipo de hardware(Eth.1)	Tipo de protocolo(IP:0x00)		
28 bytes	HLEN	PLEN	Operación	
			Sender HA(0-3)	
			Sender IP(4-5)	Sender IP(0-1)
			Sender IP(2-3)	Target HA(0-1)
			Target HA(2-3)	
			Target IP (0-3)	

- Tipo de hardware: especifica el tipo de interfaz de hardware para el que se busca respuesta.
- Tipo de protocolo: especifica el tipo de dirección de alto nivel proporcionada por el transmisor.
- HLEN: especifica la longitud de la dirección de hardware (Ethernet 6 octetos).
- PLEN: especifica la longitud de la dirección del protocolo de alto nivel (IP 4 octetos).
- Operación: 1 para solicitud ARP, 2 para respuesta ARP.
- Sender HA: contiene la dirección física del transmisor.

- Sender IP: contiene la dirección IP del transmisor.

- Target AA: contiene la dirección física del objetivo.

Target IP: contiene la dirección IP del objetivo.

La máquina objetivo completa las direcciones faltantes, volteo los puertos de objetivo y trasmisor, y cambia la operación a respuesta.

Desventajas de IP

La desventaja más obvia es que las direcciones se refieren a las conexiones de red, no a la computadora anfitrión, por lo que si ésta se mueve de una red a otra, su dirección IP debe cambiar.

Otra debilidad del esquema de direccionamiento en Internet es que cuando una red tipo C crece, hasta tener más de 255 hosts, tiene que cambiar su dirección a una tipo B. Aunque esto puede parecer un problema menor, el cambio de direcciones de red puede tomar demasiado tiempo y ser difícil de dejar.

Consideremos un host con dos conexiones a Internet, lo que significa que tiene más de una dirección IP. Como el ruteo utiliza la parte de red de la dirección IP, el camino tomado por los paquetes que viajan hacia un anfitrión con muchas direcciones IP depende de la dirección utilizada. Esto implica que deben aprenderse varias direcciones para un mismo anfitrión a fin de poder manejarlo.

IPv6

La versión 4 del protocolo IP demostró ser muy robusta, de fácil implementación e interoperabilidad, no obstante lo cual el protocolo original no previó algunos aspectos tales como el crecimiento de las redes y el posible agotamiento de las direcciones IP, el crecimiento de las tablas de ruteo, los problemas relacionados con la seguridad de los datos transmitidos o la prioridad en la entrega de determinados tipos de paquetes.

Las especificaciones de IPv4 reservan 32 bits para el rangoamiento, permitiendo generar más de 4 mil millones de direcciones diferentes, que se dividieron en tres clases de tamaño fijo: A, B y C. Aunque la intención de la división era flexibilizar la distribución de direcciones abarcando redes de diferentes tamaños, este tipo de clasificación demostró ser ineficiente. Así, la clase A atendía en número muy pequeño de redes pero ocupaba la mitad de las direcciones; para direccionar 300 dispositivos en una red era necesario obtener un bloq. que clase B (despidiendo casi 65000 direcciones); y las 256 direcciones de una red clase C no satisfacían las necesidades de la mayoría de las redes.

Otro factor que contribuyó al desperdicio de direcciones fue el hecho de que decenas de rangos clase A fueron asignados íntegramente a grandes organizaciones, poniendo a disposición de cada una

4 IPv6

- Sender IP: contiene la dirección IP del transmisor.

- Target AA: contiene la dirección física del objetivo.

Target IP: contiene la dirección IP del objetivo.

La máquina objetivo completa las direcciones faltantes, volteo los puertos de objetivo y trasmisor, y cambia la operación a respuesta.

Desventajas de IP

La desventaja más obvia es que las direcciones se refieren a las conexiones de red, no a la computadora anfitrión, por lo que si ésta se mueve de una red a otra, su dirección IP debe cambiar.

Otra debilidad del esquema de direccionamiento en Internet es que cuando una red tipo C crece, hasta tener más de 255 hosts, tiene que cambiar su dirección a una tipo B. Aunque esto puede parecer un problema menor, el cambio de direcciones de red puede tomar demasiado tiempo y ser difícil de dejar.

Consideremos un host con dos conexiones a Internet, lo que significa que tiene más de una dirección IP. Como el ruteo utiliza la parte de red de la dirección IP, el camino tomado por los paquetes que viajan hacia un anfitrión con muchas direcciones IP depende de la dirección utilizada. Esto implica que deben aprenderse varias direcciones para un mismo anfitrión a fin de poder manejarlo.

IPv6

La versión 4 del protocolo IP demostró ser muy robusta, de fácil implementación e interoperabilidad, no obstante lo cual el protocolo original no previó algunos aspectos tales como el crecimiento de las redes y el posible agotamiento de las direcciones IP, el crecimiento de las tablas de ruteo, los problemas relacionados con la seguridad de los datos transmitidos o la prioridad en la entrega de determinados tipos de paquetes.

Las especificaciones de IPv4 reservan 32 bits para el rangoamiento, permitiendo generar más de 4 mil millones de direcciones diferentes, que se dividieron en tres clases de tamaño fijo: A, B y C. Aunque la intención de la división era flexibilizar la distribución de direcciones abarcando redes de diferentes tamaños, este tipo de clasificación demostró ser ineficiente. Así, la clase A atendía en número muy pequeño de redes pero ocupaba la mitad de las direcciones; para direccionar 300 dispositivos en una red era necesario obtener un bloq. que clase B (despidiendo casi 65000 direcciones); y las 256 direcciones de una red clase C no satisfacían las necesidades de la mayoría de las redes.

Otro factor que contribuyó al desperdicio de direcciones fue el hecho de que decenas de rangos clase A fueron asignados íntegramente a grandes organizaciones, poniendo a disposición de cada una

de ellas 16.777.216 direcciones. Además, se reservaron 35 rangos de direcciones clase A para usos específicos tales como multicast, broadcast y uso futuro.

Soluciones paliativas

Se comenzaron a debatir estas fechas para resolver el agotamiento de las direcciones IP y el crecimiento de las tablas de ruteo. La IETF creó el grupo ROAD en 1991.

CIDR

Propuesto por ROAD, las ideas básicas detrás del CIDR (Classless Inter-domain Routing) eran poner fin al uso de clases de direcciones para permitir la distribución de bloques de tamaño adecuado a las necesidades reales de cada red, y la agregación de rutas para reducir el tamaño de la tabla de ruteo. Los bloques CIDR se identifican mediante prefijos, donde una dirección tiene la forma a.b.c.d/x.

DHCP

Otra solución fue el protocolo DHCP (Dynamic Host Configuration Protocol). A través de DHCP, un host puede obtener una dirección IP automáticamente y adquirir información adicional (máscara de subred, dirección del router por defecto, dirección del servidor DNS local). DHCP ha sido muy utilizado por los ISP debido a que les permite asignar direcciones IP temporarias a sus clientes conectados. Un servidor DHCP tendrá una lista de direcciones IP disponibles; cada vez que un nuevo cliente se conecte a la red le será asignada una de estas direcciones en forma aleatoria, que luego será devuelta cuando el cliente se desconecte.

NAT

NAT (Network Address Translation) fue otra técnica paliativa para resolver el problema del agotamiento de las direcciones IPv4. La idea básica es permitir que varios hosts puedan salir a Internet con una única dirección IP o con un número pequeño de ellas. Dentro de una red, cada equipo recibe una dirección IP privada única que es utilizada para rutear el tráfico interno. Sin embargo, cuando un paquete debe ser enrutado fuera de la red, las direcciones IP privadas se traducen a direcciones públicas globalmente únicas.

Para esto se definieron los 3 rangos de direcciones IP privadas vistos anteriormente siendo la única regla de utilización que ningún paquete que contiene estas direcciones puede atravesar la Internet pública.

El uso de NAT demostró ser eficiente en cuanto a la economía de direcciones IP, además de presentar algunos otros aspectos positivos tales como facilitar la numeración interna de las redes, ocultar la topología

de las redes, y sólo permite la entrada de paquetes generados en respuesta a una solicitud de la red. Pero NAT presenta inconvenientes que no compensan las ventajas que ofrece.

NAT rompe con el modelo end-to-end de Internet ya que no permite conexiones directas entre dos hosts, lo que dificulta el funcionamiento de aplicaciones como P2P, VoIP o VPN. Otro problema es la baja escalabilidad, ya que el número de conexiones simultáneas es limitado y además requiere una gran capacidad de procesamiento por parte del dispositivo traductor. El uso de NAT también impide trazar el camino del paquete (traceroute) y dificulta la utilización de algunas técnicas de seguridad como IPsec. Además, su uso genera una falsa sensación de seguridad ya que a pesar de no permitir la entrada de paquetes no autorizados, NAT no realiza ningún tipo de filtro ni verificación de los paquetes que lo atraviesan.

Aunque esas soluciones han disminuido la demanda de direcciones IP, no han sido suficientes para resolver los problemas derivados del crecimiento de Internet. Pero estas medidas permitieron que hubiera más tiempo para desarrollar una nueva versión del protocolo IP, que se basara en los principios que contribuyeron al éxito de IPv4 pero que también fuese capaz de superar las fallas que se detectaron.

Se creó un grupo de trabajo llamado IPhg, buscando crear un protocolo que contara con escalabilidad, seguridad, configuración y administración de redes, soporte para QoS, movilidad, políticas de ruta, transición. Luego de varios proyectos, la nueva versión de IP pasó a llamarse IPv6. Entre los principales cambios se destacan:

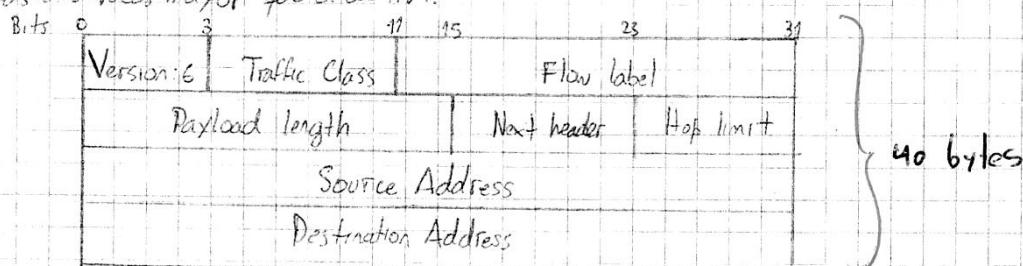
- Menor espacio de direccionamiento: en IPv6 el espacio de direccionamiento aumentó de 32 a 128 bits, permitiendo niveles más específicos de agregación de direcciones, identificar una cantidad mucho mayor de dispositivos en la red e implementar mecanismos de autoconfiguración. Se mejoró la escalabilidad del router multicast mediante la adición del campo alcance en la dirección multicast. También se definieron las direcciones anónimas.
- Simplificación del formato de encabezado: algunos campos de IPv4 se eliminaron o se convirtieron en operaciones, para reducir el costo de procesamiento de los paquetes en los routers.
- Soporte para encabezados de extensión: las opciones no son parte ya del encabezado base, permitiendo un ruteo más eficiente y menos rigurosos en cuanto al tamaño y la cantidad de opciones, y mayor flexibilidad para las opciones en un futuro.
- Capacidad de identificar flujos de datos: se añegó un nuevo recurso que permite identificar paquetes que pertenezcan a determinados flujos de tráfico que pueden requerir tratamientos especiales.
- Soporte para autenticación y privacidad: se especificaron encabezados de extensión capaces de pro-

ver mecanismos de autenticación y garantizar la integridad y confidencialidad de los datos transmitidos.

Además, IPv6 también modificó el tratamiento de la fragmentación de los paquetes que pasó a ser realizada sólo en el origen permitió el uso de conexiones end-to-end y aportó recursos que facilitan la configuración de redes, entre otros aspectos mejorados.

Encabezado de IPv6

Se realizaron algunos cambios en el formato del encabezado base de IPv6 para volverlo más simple (si lo 8 campos y un tamaño fijo de 40 bytes), más flexible y más eficiente, previendo su extensión por medio de encabezados adicionales que no necesitan ser procesados por todos los routers intermedios. Estos cambios permitirían que, incluso con 128 bits para direccionamiento, el tamaño total del encabezado de IPv6 sea apenas dos veces mayor que el de IPv4.



- Versión: identifica la versión del protocolo utilizado.
- Clase de tráfico: identifica y diferencia los paquetes por clases de servicios o prioridad. Similar a TOS de IPv4.
- Identificador de flujo: identifica y diferencia paquetes del mismo flujo en la capa de red. Este campo permite que el router identifique el tipo de flujo de cada paquete, sin necesidad de verificar su aplicación.
- Tamaño de los datos: indica el tamaño de los datos enviados junto con el encabezado de IPv6. En el cálculo del tamaño también se incluyen los encabezados de extensión.
- Siguiente encabezado: identifica el encabezado que sigue al encabezado de IPv6. Este campo no solo contiene valores referentes a otros protocolos sino que también indica los valores de los encabezados de extensión.
- Límite de salto: indica el número máximo de routers que el paquete IPv6 puede pasar antes de ser descartado, se decremente en cada salto.
- Dirección de origen: indica la dirección de origen del paquete.
- Dirección de destino: indica la dirección de destino del paquete.

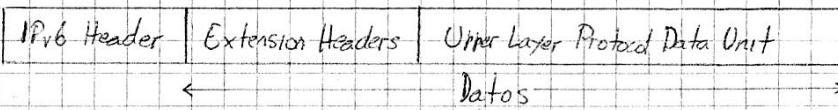
Entre los cambios se destaca la eliminación de 6 campos del encabezado de IPv4 debido a que son fun-

eladas ya no son necesarias o son implementadas por las cabeceras de extensión. Como las opciones ahora forman parte de los encabezados de extensión, se pudieron eliminar los campos Opciones y Padding. El campo Header Size también se eliminó, ya que el tamaño del encabezado de IPv6 es fijo. Los campos Identificación, Flags y Fragment Offset se eliminaron, ya que los datos referentes a la fragmentación ahora se indican en un encabezado de extensión apropiado. Para aumentar la velocidad de procesamiento de los routers se eliminó el campo Header Checksum, ya que este cálculo es realizado por los protocolos de las capas superiores.

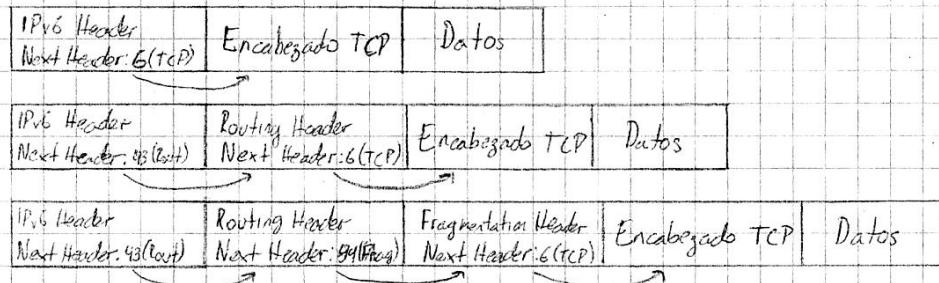
Los campos Tipo de Servicio, Tamaño total, TTL y Protocolo cambiaron de nombre y posición para facilitar el procesamiento de estos datos por parte de los routers. También se agregó el campo de Identificador de flujo, suministrado al protocolo IP otro mecanismo de soporte para QoS. Los campos Versión, Dirección de Origen y Dirección de Destino se mantuvieron, modificando el tamaño de los dos últimos.

Encabezados de extensin

El esquema general de un datagrama IP6 es el siguiente:



En IPv6 las opciones se tratan por medio de los encabezados de extensión. Éstos se encuentran entre el encabezado base y el encabezado de la capa de transporte. Estos encabezados no tienen ni cantidad ni tamaño fijo. Un esquema de ejemplo podría ser:



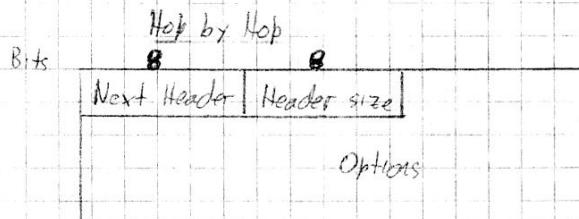
Como se ve, si en un mismo paquete existen múltiples encabezados de extensión, éstos serán agregados en serie formando una cadena de encabezados. Las especificaciones de IPv6 definen 6 encabezados de extensión: Hop by Hop, Destination, Routing, Fragmentation, Authentication y Encapsulating Security Payload.

El uso de encabezados de extensión busca aumentar la velocidad de procesamiento en los routers, ya que el único encabezado procesado en ellos es el de Hop by Hop. Además, se pueden definir y utilizar nuevos encabezados de extensión sin tener que modificar el encabezado base.

Es importante destacar que, para evitar que los nodos existentes a lo largo del camino del paquete tengan que recorrer toda la cadena de encabezados de extensión para saber cuáles datos tratar, estos encabezados

se deben enviar respetando un orden determinado. En general, los encabezados importantes para todos los nodos involucrados en el ruta se deben colocar en primer lugar y los que solo son importantes para el destino final se colocan al final de la cadena. La ventaja de esto es que el nodo puede detener el procesamiento de los encabezados si no encuentra algún encabezado de extensión dedicado al destino final teniendo la certeza de que no hay más encabezados importantes. Así se puede mejorar significativamente el procesamiento de paquetes. Por lo tanto, la secuencia a seguir sería Hop by Hop, Routing, Fragmentation, Authentication, Encapsulation Security Payload, Destination.

También vale la pena observar que si un paquete fue enviado a una dirección multicast, los encabezados de extensión serán examinados por todos los nodos del grupo.



Identificando por el valor 0 en el campo Next Header, el encabezado de extensión Hop by Hop se debe colocar inmediatamente después del encabezado base de IPv6. Los datos transmitidos por el mismo deben ser examinados por todos los nodos intermedios a lo largo del camino del paquete hasta que llega a su destino. En su ausencia, el router sabe que no necesita procesar ningún dato adicional y puede encaminar el paquete hacia el destino final inmediatamente.

- Next header: identifica el tipo de encabezado que sigue al Hop by Hop.

- Header size: indica el tamaño del encabezado Hop by Hop en unidades de 8 bytes, excluyendo los 8 primeros.

Options: contiene una o más opciones y su tamaño es variable. Los dos primeros bits codifican qué hacer en caso de que el nodo procesando la información no reconozca la opción:

• 00: ignorar y continuar el procesamiento.

• 01: descartar el paquete.

• 10: descartar el paquete y enviar un mensaje ICMP Parameter Problem a la dirección de origen del paquete.

• 11: idem anterior, pero solo si el destino no es una dirección multicast.

El tercer bit especifica si la información opcional puede cambiar en ruta (1) o no (0).

Hasta el momento existen dos tipos definidos para el encabezado Hop by Hop: Router Alert/Advertiser.

para informar a los intermediarios que el mensaje a ser encadenado exige tratamientos especiales, usada en MLD y PSVP) y Jumbofram (utilizada para informar que el tamaño del paquete IPv6 es mayor a 64 kb).

Destination

Identificado por el valor 60 en el campo Next Header, el encabezado de extensión Destination transporta datos que deben ser procesados por el nodo de destino del paquete. La definición de sus campos es igual a la de los encabezados Hop by Hop.

Este encabezado se utiliza en el soporte para movilidad en IPv6 a través de la opción Home Address, la cual contiene la dirección de origen del nodo móvil cuando está en tránsito.

Routing

Bits	8	8	8	8
Next header	Header size	Routing Type	Segments Left	
		Reserved		
		Dirección de origen		
		Dirección 1 (Tipo 0)		
		...		

Identificado por el valor 43 en el campo Next Header, el encabezado de extensión Routing fue inicialmente desarrollado para listar uno o más nodos intermedios a ser visitados por el paquete antes de llegar a su destino. Esta función, realizada por el encabezado Routing Type 0, se volvió obsoleta debido a problemas de seguridad. Se definió un nuevo encabezado Routing Type 2, para ser utilizado como parte del mecanismo de soporte para movilidad en IPv6.

- Routing Type: 0, 1 o 2.

- Segments left: en tipo 0, indica el número de saltos a ser realizados antes de que el paquete llegue a su destino final.

Fragmentation

Bits	8	8	13	2	1
Next Header	Reservado	Fragment Offset	Res	M	
		Identification			

Identificado por el valor 44 en el campo Next Header, el encabezado de extensión Fragmentation se utiliza cuando el paquete IPv6 a ser enviado es mayor que el Path MTU.

- Fragment offset: indica en unidades de 8 bytes la posición de los datos transportados por el fragmento actual respecto del inicio del paquete original.

- M: si está marcado con 1, indica que hay más fragmentos.

- Identificación: valor único generado por el nodo de origen para identificar el paquete original. Se utiliza para detectar los fragmentos de un mismo paquete.

Authentication / Encapsulating Security Payload

Los encabezados de extensión Authentication y Encapsulating Security Protocol, indicados respectivamente por los valores 51 y 52 en el campo Next Header, forman parte del encabezado IPsec. Aunque las funcionalidades de IPsec son idénticas tanto en IPv4 como en IPv6, su utilización con IPv6 es facilitada por el hecho de que sus principales elementos forman parte integral de la nueva versión de IP.

Direccionamiento IPv6

En IPv4, el campo del encabezado reservado para direccionamiento tiene 32 bits, lo que permite un máximo de 4.294.967.296 direcciones diferentes. En la época de su desarrollo, esta cantidad se consideraba suficiente para identificar todas las computadoras en la red y soportar el surgimiento de nuevas redes.

IPv6 tiene un espacio de direccionamiento de 128 bits, aproximadamente 340 sextillones de direcciones posibles.

La representación de las direcciones IPv6 divide la dirección en 8 grupos de 16 bits, separados mediante colonas con dígitos hexadecimales (pueden usarse tanto mayúsculas como minúsculas). Además, se pueden aplicar reglas de abreviatura para facilitar la escritura de algunas direcciones muy extensas. Se permite omitir los 0s a la izquierda de cada bloque de 16 bits y también reemplazar una larga secuencia de 0s por :: (solo una vez para evitar ambigüedades).

Otra representación importante es la de los prefijos de red. En las direcciones IPv6 continúa escribiéndose del mismo modo que en IPv4, utilizando la notación CIDR. Esta notación se representa con la forma dirección/tamaño del prefijo, donde este último es un valor decimal que especifica la cantidad de bits contiguos a la izquierda de la dirección que componen el prefijo.

Esta representación también permite agregar las direcciones en forma jerárquica de esta manera disminuyendo el tamaño de la tabla de rutas y agilizando el encaminamiento de los paquetes.

En IPv6 se definieron tres tipos de direcciones:

- Unicast: identifican una única interfaz, de modo que un paquete enviado a una dirección Unicast se entrega a una interfaz. Comunicación uno a uno.

Anycast: identifican un conjunto de interfaces. Un paquete enviado a una dirección anycast se entrega a la interfaz perteneciente a este conjunto más próxima al origen. Comunicación de uno a "uno de muchos".

Multicast: también identifican un conjunto de interfaces, pero un paquete enviado a una dirección multicast se entrega a todos las interfaces asociadas a esa dirección. Comunicación de uno a muchos.

En IPv6 no existe la dirección broadcast responsable de dirigir un paquete a todos los nodos de un mismo dominio; esta función se asignó a determinados tipos de direcciones multicast.

Direcciones unicast

Las direcciones unicast se utilizan para comunicaciones entre dos nodos, y su estructura fue definida para permitir agrupaciones con prefijos de tamaño flexible. Existen 3 tipos ademas de direcciones especiales.

Global Unicast

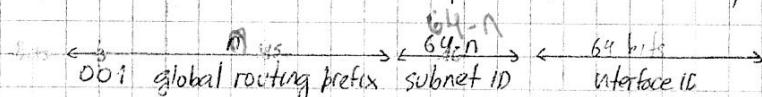
IPv4 Pública

Equivalente a las direcciones IPv4 públicas, las direcciones global unicast son globalmente ruteables y accesibles en la Internet IPv6. Está formada por tres partes: el prefijo de encaminamiento global (usado para identificar el tamaño del bloque atribuido a una red), la identificación de la subred (usada para identificar un enlace en una red) y la identificación de la interfaz (debe identificar de forma única una interfaz dentro de un enlace).

Su estructurada fue proyectada para utilizar los 64 bits más hacia la izquierda para identificar la red y los 64 bits más hacia la derecha para identificar la interfaz. Por lo tanto, salvo en casos específicos, se tienen 2^{64} dispositivos por subred.

Actualmente está reservado el rango 2000::/3, o sea de 2000:: a 3FFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF.

Se pueden tener 65536 subredes con un bloque 148 y 256 con un bloque 156.



Link Local

IPv6 privada sólo enlace

Pudiendo utilizarse sólo en el enlace específico en el cual la interfaz está conectada, la dirección link local es asignada automáticamente usando el prefijo FE80::/64. Vale la pena destacar que los routers no deben encaminar paquetes cuyo origen o destino sea una dirección link local hacia otros enlaces.

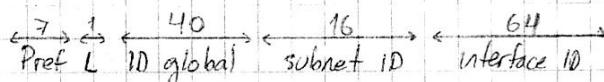


Unique Local Address

Dirección con grandes probabilidades de ser globalmente única, utilizada solamente para comunicaciones locales, generalmente dentro de un mismo enlace o conjunto de enlaces. Una dirección ULA no debe ser ruteable en

la Internet global, y está formada por el prefijo Fc00::/7, una banda L. (si vale 1(Fc) el prefijo es atribuido localmente, si es 0(Fc) lo asigna una autoridad central), un identificador global (señalado anterior y el ID de la interfaz).

Su utilización permite que cualquier enlace posea un prefijo /48 privado y globalmente único. Por lo tanto, en el caso que se interconecten dos redes, es probable que no haya conflicto de direcciones ni necesidad de renombrar la interfaz. Además, la dirección URA es independiente del proveedor, pudiendo ser utilizado en la reconstrucción del ancho dentro aunque no haya conexión a Internet. Otra ventaja es que su prefijo se puede bloquear fácilmente en caso de un rusho accidental y así no habrá conflicto con otras direcciones.



Identificador de interfaz

Los identificadores de interfaz (IID), utilizados para distinguir las interfaces dentro de un enlace, deben ser únicos dentro del mismo prefijo de subred. El mismo IID se puede usar en múltiples interfaces de un único nodo, pero éstas deben estar asociadas a subredes diferentes.

Normalmente se utiliza un IID de 64 bits, el cual se puede obtener de diferentes maneras: se puede configurar manualmente, a partir del mecanismo de autoconfiguración stateless de IPv6, o a partir de servidores DHCPv6 (stateful), o se pueden formar a partir de una clave pública (CGA). Se recomienda construir el IID en base a la dirección MAC de la interfaz, en el formato EUI-64.

Un IID basado en el formato EUI-64 se genera de la siguiente manera: si la interfaz tiene una dirección MAC de 64 bits (estándar EUI-64), sólo debe complementar el séptimo bit más a la izquierda (llamado bit U/L) de la dirección MAC. Si la interfaz utiliza una dirección MAC de 48 bits (estándar IEEE 802), primero se agregan los dígitos hexadecimales FFFE entre el tercer y cuarto bit de la dirección MAC y luego se complementa el bit U/L.

Direcciones especiales

Existen algunas direcciones IPv6 especiales utilizadas para fines específicos:

- Dirección no especificada: representada mediante la dirección ::/128, esta dirección nunca debe ser atribuida a ningún nodo, ya que sólo indica la ausencia de una dirección (unión que se la puede usar en proceso de intercambio como dir. de origen). La dirección unespecified no debe ser utilizada como dir. de destino de paquetes IPv6.
- Dirección localhost: representada mediante la dirección unicast ::/128, esta dirección se utiliza para referenciar la propia máquina y es muy utilizada para realizar pruebas internas. Este tipo de dirección no debe atribuirse.

larse a ninguna interfaz física. No debe ser utilizada como dir. de origen en paquetes IPv6 enviados a otros nodos.

Aemás, un paquete IPv6 con una dirección loopback como destino no puede ser enviado por un router IPv6, y si un paquete recibido en una interfaz tiene una dir. loopback como destino, debe ser descartado.

- Direcciones multihomed IPv4: Representadas mediante ::FFFF:wxyz, se utiliza para mapear una dirección IPv4 en una dirección IPv6 de 128 bits, donde wxyz representa los 32 bits de la dirección IPv4, utilizando dígitos decimales. Se aplica en técnicas de transición para que se comuniquen nodos IPv6 e IPv4.

Direcciones anycast

de uno a uno de muchos

a un grupo responde uno solo

Una dirección IPv6 anycast se utiliza para identificar un grupo de interfaces, aunque con la propiedad de que un paquete enviado a una dirección anycast es encaminado solamente a la interfaz del grupo más próximo al origen del paquete.

Las direcciones anycast se distribuyen a partir del rango de direcciones unicast y no hay diferencias sintácticas entre las mismas. Por lo tanto una dirección unicast atribuida a más de una interfaz se transforma en una dirección anycast (en tal caso se deben configurar explícitamente los nodos para que lo sepan). En los routers, esta dirección debe ser configurada como una entrada independiente (prefijo 1128).

Este esquema de direccionamiento se puede usar para descubrir servicios en la red, como servidores DNS y proxies HTTP. También se puede utilizar para realizar balanceo de carga en situaciones donde múltiples nodos proveen el mismo servicio, para localizar routers que provean acceso a una determinada subred, o para localizar los agentes de origen en redes con soporte para movilidad IPv6.

Todos los routers deben soportar la dirección anycast Subnet-Router, formada por el prefijo de la subred y el IID en 0.0.0.1. Un paquete enviado a esta dirección será entregado al router más próximo al origen dentro de la subred.

Direcciones multicast

en todo un grupo

Las direcciones multicast se utilizan para identificar grupos de interfaces. Los paquetes enviados a esas direcciones se entregan a todas las interfaces que componen el grupo. En IPv4 el soporte multicast es opcional (fue introducido como una extensión), pero en IPv6 se requiere que todos los nodos soporten multicast.

Su funcionamiento es similar al de broadcast, dado que un único paquete es enviado a varios hosts, con la diferencia que en broadcast el paquete se envía a todos los hosts de la red y en multicast sólo a un grupo de hosts. De este modo, la posibilidad de transmitir solo una copia de los datos a todos los elementos del grupo puede reducir la utilización de recursos en una red, además de optimizar la entrega de datos a hosts receptores.

Los direcciones multicast no deben ser utilizadas como dirección de origen de un paquete. Es las direcciones

derivan del bloque FF00: 18, donde el prefijo FF, que identifica una dirección multicast, es precedido por 4 bits de banderas, y un valor de 4 bits que define el alcance de la dir. multicast. Los 112 bits restantes se utilizan para identificar el grupo multicast.

Flags

- El primer bit más a la izquierda está reservado y debe ser 0.
- Flag R: si el valor es 1 indica que la dirección multicast lleva la dirección de un punto de encuentro.
- Flag P: si el valor es 1 indica que la dirección multicast está basada en un prefijo de red.
- Flag T: si el valor es 0 indica que la dirección multicast es permanente (atribuida por la IANA). Si es 1, indica que ha sido atribuida dinámicamente.

Scope

Estos 4 bits se utilizan para delimitar el área que abarca un grupo multicast. Puede ser: 1 (interfaz local), 2 (nodos de un enlace), 3 (nodos de una subred), 4 (menor área configurada manualmente), 5 (nodos de un sitio), 8 (varios sitios de una organización), E (toda la Internet). O y F están reservados, mientras que 6, 7, 9, A, B, C y D no están distribuidos.

Direcciones multicast permanentes

- FF01: 1: alcance interfaz, todos los interfaces en un nodo.
- FF02: 2: alcance interfaz, todos los routers en un nodo.
- FF02: 1: alcance enlace, todos los nodos del enlace. (broadcast?)
- FF02: 2: alcance enlace, todos los routers del enlace.
- FF02: 5: alcance enlace, routers OSPF.
- FF02: 6: alcance enlace, routers OSPF designados.
- FF02: 9: alcance enlace, routers RIP.
- FF02: D: alcance enlace, routers PIM.
- FF02: 1.2: alcance enlace, agentes DHCP
- FF02: 1. FFXX:XXXX: alcance enlace, solicited-node.
- FF05: 2: alcance sitio, todos los routers en un sitio.
- FF05: 1.3: alcance sitio, servidores DHCP en un sitio.
- FF05: 1.4: alcance sitio, agentes DHCP en un sitio.

- FF0X::101: alcance variado, Network Time Protocol.

La dirección multicast solicitado-node identifica un grupo multicast del cual todos los nodos pertenecen a formar parte una vez que les es asignada una dirección unicast o anycast y se forma agregando al prefijo FF02::1.FF00::104 los 24 bits más a la derecha del 110, y para cada dirección unicast o anycast del nodo existe una dirección multicast solicitado-node correspondiente. Esta dirección es utilizada por el protocolo de Descubrimiento de Vecinos.

Multicast a partir de unicast

Con el propósito de reducir el número de protocolos necesarios para la distribución de direcciones multicast, se definió un formato extendido de dirección multicast, que permite distribuir direcciones en base a prefijos unicast. En estas direcciones, el flag P se marca con el valor 1, el alcance no importa (an que no debe exceder el alcance del prefijo unicast), luego vienen 8 bits reservados en 0, 8 bits que especifican el tamaño del prefijo de red indicado en los siguientes 64 bits (si es menor a 64, los bits no utilizados se marcan en 0) y luego el campo identificador del grupo. Debe notarse que si P es 1, T debe ser 1, ya que no representa una dirección de la IANA.

Bits <8> < 4 > < 4 > < 8 > < 8 > < 64 > < 32 >
FF OR11 Scope Reservado Prefix size Prefix de la red group ID

Direccionamiento

Al igual que en IPv4, las direcciones IPv6 se atribuyen a las interfaces físicas, no a los nodos, de modo que cada interfaz necesita al menos una dirección unicast. Sin embargo, se puede atribuir a una única interfaz múltiples direcciones IPv6, independientemente del tipo (unicast, anycast, multicast) o subtipo (loopback, link local, etc.). Así, un nodo se puede identificar a través de cualquier dirección de sus interfaces, y por lo tanto se hace necesario elegir entre sus múltiples direcciones cuáles se utilizarán como direcciones de origen y destino al establecer una conexión.

ICMPv6

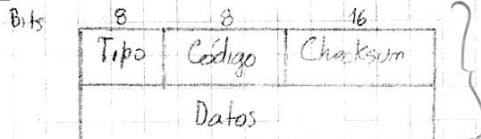
ICMPv6 tiene las mismas funciones (informar errores en el procesamiento de paquetes, enviar mensajes sobre el estado o las características de la red) que ICMPv4, pero no es compatible con éste y presenta un mayor número de mensajes y funcionalidades. Ahora ICMPv6 es responsable de realizar las funciones de los protocolos ARP e IGMP. El valor del campo Next Header para ICMPv6 es 58 y su implementación es obligatoria.

Header IPv6	Cabeceras de extensión	ICMPv6
-------------	------------------------	--------

En un paquete IPv6, el ICMPv6 se coloca inmediatamente después del encabezado base de IPv6 y los encabezados de extensión si los hubiera.

ICMPv6 es un protocolo clave en la arquitectura IPv6, ya que además de gestionar los grupos multicast (MLD))

y de la resolución de direcciones, sus mensajes son esenciales para el funcionamiento del protocolo de Descubrimiento de Vecinos, para el soporte a la movilidad y para el proceso de descubrimiento de la menor MTU en un enlace.



- TIPO: especifica el tipo de mensaje, lo que determina el formato del cuerpo del mensaje.

- Código: ofrece algunos datos adicionales para determinados tipos de mensajes.

- Checksum: utilizada para detectar datos corruptos.

- Datos: presenta información del diagnóstico y error según el tipo de mensaje. Para ayudar en la resolución de problemas, los mensajes de error incluyen en este campo el paquete que invocó el mensaje.

Los mensajes de error ICMPv6 pueden ser: Destination Unreachable (tipo 1, indica fallas en la entrega del paquete o problemas en la comunicación), Packet Too Big (tipo 2, indica que el tamaño del paquete es mayor que la MTU de un enlace), Time Exceeded (tipo 3, indica que el Hop Limit o el tiempo de ensamblaje del paquete fue excedido), Parameter Problem (tipo 4, indica un error en alguno de los campos del encabezado IPv6 o el campo Next Header).

Los mensajes de información pueden ser: Echo Request/Reply (tipo 128/129, usado por el comando ping), Multicast Listener Query/Report/Done (tipo 130/131/132, utilizados en la gestión de grupos multicast), Router/Neighbor Solicitation/Advertisement y Redirect Message (tipo 133/134/135/136/137, utilizados en Descubrimiento de Vecinos), etc.

Descubrimiento de vecinos

El protocolo de Descubrimiento de Vecinos agiliza algunos procesos de configuración de red con respecto a IPv4 combinando las funciones de protocolos como ARP, ICMP Router Discovery y ICMP Redirect, y además agrega nuevos métodos.

Este protocolo es utilizado por los hosts y routers para determinar la dirección MAC de los nodos de la red, encontrar routers vecinos, determinar prefijos y otros datos de configuración de la red, detectar direcciones duplicadas, determinar la accesibilidad de los routers, redirecciones paquetes, autoconfigurar direcciones, etc.

Los mensajes Neighbor Discovery se configuran con un Hop Limit de 255 para asegurar que los mensajes recibidos tengan su origen en un nodo del mismo enlace, descartando los mensajes que tengan valores diferentes.

Descubrimiento de MACs

Esta funcionalidad se utiliza para determinar la dirección MAC de los vecinos del mismo enlace: un host envía un mensaje Neighbor Solicitation a la dirección multicast solicitando al vecino informando su dirección

MAC. Al recibir el mensaje, el vecino responde enviando un mensaje Neighbor Advertisement informando su dirección MAC.

Descubrimiento de routers y prefijos

Esta funcionalidad se utiliza para localizar routers vecinos dentro del mismo enlace, así como para aprender prefijos y parámetros relacionados con la autoconfiguración de direcciones. Esta información se envía desde un router local, a través de mensajes Router Advertisement encaminados a la dirección multicast all-routers, scope link local $\text{FF02}::1$.

Autoconfiguración

Es un mecanismo seguido por un host para autoconfigurar interfaces IPv6. El tipo stateful es provista por un servidor de direcciones (DHCPv6); se utiliza cuando no se encuentra ningún router o cuando se indica en mensajes Router Advertisement. Este mecanismo puede proveer direcciones IPv6 y diferentes parámetros de la red (DNS, NTP, SIP, etc.).

El mecanismo de autoconfiguración stateless permite atribuir direcciones IPv6 a las interfaces a través realizando una configuración mínima de los routers. Para generar una dirección IP un host utiliza una combinación de datos locales, como la dirección MAC de la interfaz o un valor aleatorio para generar el ID, e información recibida de los routers, como múltiples prefijos. Si no hay routers presentes, el host genera sólo la dirección link-local con el prefijo $\text{FE80}::$. Los routers sólo usan este mecanismo para generar direcciones link-local.

El mecanismo de autoconfiguración de direcciones se ejecuta respetando los siguientes pasos: se genera una dirección link-local agregando el prefijo $\text{FE80}::/64$ al IID; esta dirección pasa a formar parte de los grupos multicast solicited-node y all-node. Luego se verifica la unicidad de la dirección link-local generada: si ya está siendo utilizada el proceso se interrumpe y es necesario configurar manualmente; si en cambio es única y válida se la asigna a la interfaz. Despues, el host envía un mensaje Router Solicitation al grupo multicast all-routers. Todos los routers del enlace responden con un mensaje RA informando los routers por defecto, el valor predeterminado para Hop Limit, la MTU del enlace, la lista de prefijos (para los cuales se generarán direcciones automáticamente), etc.

Path MTU Discovery Maximum Transmission Unit

Dependiendo de los protocolos de routing cada enlace de la red puede tener un valor de MTU diferente. Para poder enviar paquetes mayores que la MTU del enlace, éstos se deben fragmentar en paquetes menores que serán ensamblados al llegar a su destino.

En IPv6 la fragmentación de paquetes se realiza sólo en el origen, no estando permitida en los routers intermedios. Este proceso tiene por objetivo reducir el overhead del cálculo de los encabezados modificados en los routers intermedios.

Dura ello, en el inicio del proceso de fragmentación se utiliza el protocolo Path MTU Discovery, que desobre de forma dinámica cuál es el tamaño máximo de paquetes permitido identificando previamente las MTU de cada enlace en el camino hacia el destino. Todos los routers IPv6 deben soportar el protocolo PMTUD.

El proceso comienza suponiendo que la MTU de todo el camino es igual a la MTU del primer salto. Si el tamaño de los paquetes enviados es mayor que el que sobrepasa alguno de los routers a lo largo del camino, este lo descartará y devolverá un mensaje **ICMPv6 packet too big**, que junto con el mensaje de error devuelve el valor de la MTU del enlace siguiente. Luego de recibir este mensaje el nodo de origen reduce el tamaño de los paquetes de acuerdo con la MTU indicada en el mensaje ICMPv6.

Este procedimiento finaliza cuando el tamaño del paquete es igual o menor que la MTU del camino, por lo que estos intercambios pueden ocurrir varias veces hasta encontrar la menor MTU. Si el paquete es enviado a un grupo multicast el tamaño será la menor PMTU de todo el conjunto de destinos.

Calidad de servicio

Al principio, el protocolo IP trata todos los paquetes de la misma manera, sin ninguna preferencia a la hora de encaminarlos. Esto puede tener diferentes consecuencias en aplicaciones como voz y video, que requieren transmisión y reproducción prácticamente en tiempo real, y su calidad se puede reducir debido a la pérdida de paquetes, entrega fuera de orden, retraso o variación de la señal, problemas causados por la forma en que el tráfico llega a los routers y es manipulado por los mismos.

El concepto de Quality of Service (QoS) se utiliza para los protocolos cuya función es proporcionar la transmisión de determinados tráficos con prioridad y garantía de calidad. Existen tres modelos principales: Best Effort Internet, Integrated Services y Differentiated Services.

En el modelo Best Effort, todos los paquetes son tratados de la misma manera y no se provee calidad o garantía de tráfico. Las garantías se proveen con sobre-provisionamiento.

Integrated Services

El modelo IntServ se basa en la reserva de recursos por flujo y su utilización está normalmente asociada al protocolo RSVP, que se utiliza para reservar el recurso a lo largo del camino de un flujo que requiere QoS, desde la fuente hasta el destino.

En IPv6 se utilizan los 20 bits del campo Identificador de Flujo para identificar los flujos que requie-

ten QoS (los que no pertenecen a un flujo comparten con ellosí como los hosts y routers que envían un paquete de este tipo pero no soportan QoS, que además deben ignorarlo al recibir o enmascararlo). Los paquetes de un mismo flujo deben tener la misma dirección de origen y destino, y el mismo valor en Identificador de Flujo.

La desventaja de esta arquitectura es que RSVP es complejo; se debe mantener señalización y reserva en cada router por cada flujo. Además es poco escalable.

Differentiated Services

DiffServ trabaja por medio de clases, agregando y priorizando paquetes con requisitos de QoS similares. Los paquetes DiffServ se identifican por los 8 bits del campo Class de Tráfico, con el fin de identificar y distinguir las diferentes clases o prioridades de paquetes que requieren QoS.

En DiffServ, el campo Traffic Class se divide en 2 partes: DSCP(6 bits, 3 codifican la clase y 3 detallan de la clase) y ECN(2 bits, indica congestión de un nodo). Algunas clases de servicio en DiffServ son: Expedited Forwarding o Premium (DSCP 101110, equivale a una línea dedicada; garantiza caudal, tasa de pérdidas, retraso y jitter), Assured Forwarding (asegura trato preferente pero no fija garantías; se definen 4 clases y en cada una 3 niveles de descarte de paquetes), Best Effort con prioridad (DSCP 000xxx, sin garantías, pero trato más preferente que Best Effort) y Best Effort (DSCP 000xxxx, ninguna garantía).

Los routers internos de un dominio DiffServ clasifican y gestionan las colas, mientras que los routers frontera clasifican, miden, marcan, modelan y descartan.

Las prioridades atribuidas a cada tipo de paquete se pueden definir tanto en el origen como en los routers, y también pueden ser redifinidas por los routers intermedios a lo largo del camino. En paquetes que no requieren QoS, el campo Traffic Class tiene el valor 0.

En comparación con IntServ, DiffServ no exige ninguna identificación ni gestión de los flujos, y en general es más utilizado en las redes debido a su facilidad de implementación.

Coexistencia y transición

Toda la estructura de Internet está basada en IPv4. Un cambio inmediato de protocolo es inviable debido al tamaño y la proporción que tiene esta red, por lo que la adopción de IPv6 se debe realizar de forma gradual. Inicialmente habrá un período de transición y de coexistencia entre los dos protocolos, por lo que las redes IPv4 necesitarán comunicarse con las redes IPv6 y viceversa. Para facilitar este proceso se han desarrollado algunas técnicas que buscan mantener la compatibilidad de toda la base instalada de redes IPv4 con el nuevo protocolo IPv6. Cada una de estas técnicas tiene características específicas, y se puede utilizar individualmente.

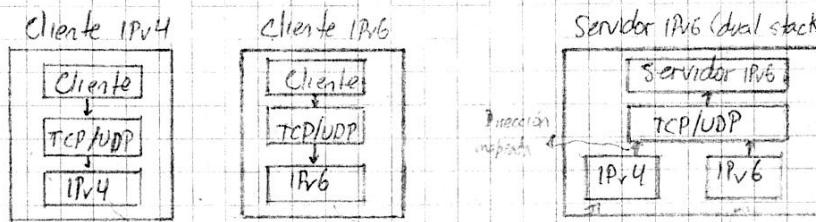
O junto con otras técnicas para adecuarse a las necesidades de cada situación.

Doble pila

En esta fase inicial de implementación de IPv6 todavía no es aceptable tener nodos que sólo soporten esta versión del protocolo IP, ya que muchos servicios y dispositivos de red continúan trabajando sólo sobre IPv4. Por esto, una posibilidad consiste en implementar el método conocido como doble pila, que permite que los hosts y routers estén equipados con pilas para ambos protocolos y tengan la capacidad de enviar y recibir ambos tipos de paquetes. Así, en la comunicación con un nodo IPv6, un nodo doble pila se comportará como un nodo sólo IPv6, y lo mismo en comunicación con un nodo IPv4.

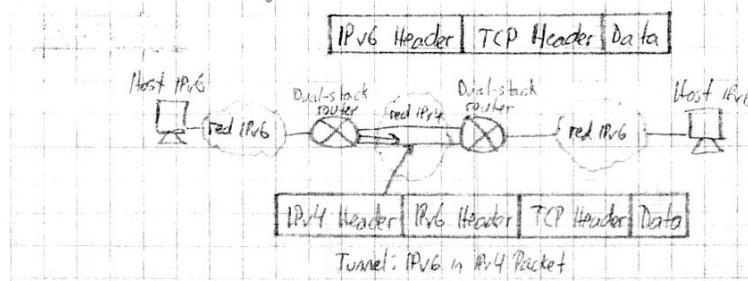
Cada nodo doble pila se configura con ambas direcciones, utilizando mecanismos IPv4 (DHCP) para adquirir su dirección IPv4 y mecanismos de protocolo IPv6 (autoconfiguración o DHCPv6) para adquirir su dirección IPv6.

Este método de transición puede facilitar la gestión de la implementación de IPv6, ya que permite implementar IPv6 en forma gradual configurando pequeñas secciones del entorno de red por vez. Además, si en un futuro se dejara de utilizar IPv4, bastaría con deshabilitar la pila IPv4 de cada nodo.



Al implementar la técnica de doble pila es importante considerar algunos aspectos, como la configuración de los servidores DNS (para que pueda resolver nombres y direcciones de ambos protocolos), la configuración de los protocolos de ruteo (que soporten ambos protocolos, como IS-IS), la configuración de los firewalls (se puede tener que usar un filtro para IPv4 y otro para IPv6) y los cambios en la administración de las redes.

Tunneling



La técnica de creación de túneles permite transmitir paquetes IPv6 a través de la infraestructura IPv4 existente.

tente sin necesidad de realizar ningún cambio en los mecanismos de ruteo, encapsulando el contenido del paquete IPv6 en un paquete IPv4.

Existen diferentes formas de encapsulamiento: paquetes IPv6 encapsulados en paquetes IPv4 (Protocolo 41, 6to4, ISATAP, Tunnel Brokers), paquetes IPv6 encapsulados en paquetes GRE (Protocolo GRE) y paquetes IPv6 encapsulados en paquetes UDP (TEREDO).

Traducción

Las técnicas de traducción permiten un enrutamiento transparente en la comunicación entre los nodos de una red IPv6 y los nodos de una red IPv4. Pueden actuar de diferentes maneras y en zonas distintas: traduciendo encabezados IPv4 en encabezados IPv6 y viceversa, convirtiendo direcciones, convirtiendo APIs de programación o actuando en el intercambio de tráfico TCP o UDP.

UDP

Hemos descripto una Internet capaz de transferir datagramas IP entre computadoras arbitrarias, donde cada datagrama se rutea a través de la red, basándose en la dirección IP de destino. Ahora ampliamos el grupo de protocolos TCP/IP al agregar un mecanismo que distingue entre muchos destinos dentro de un host, permitiendo que varios programas de aplicación que se ejecutan en una computadora envíen y reciban datagramas en forma independiente.

Puertos

Los sistemas operativos permiten que varios programas de aplicación se ejecuten al mismo tiempo. Nos referimos a cada programa en ejecución como un proceso o tarea. Puede parecer natural decir que un proceso es el destino final de un mensaje. Sin embargo, especificar que un proceso en particular en una máquina en particular es el destino final para un datagrama es un poco confuso. Primero, como los procesos se crean y se destruyen de manera dinámica, los transmisores rara vez saben lo suficiente para identificar un proceso en otra máquina. Segundo, sería bueno poder reemplazar los procesos que reciben datagramas, sin tener que informar a los transmisores (ej. reinicio). Tercero, necesitamos identificar los destinos de las funciones implementadas sin conocer el proceso que implementa la función. También es importante saber, cuando un proceso maneja varias funciones, cuál debe decidir el proceso exactamente la función deseada por el transmisor.

En vez de pensar en un proceso como destino final, imaginemos que cada máquina contiene un grupo de puntos abstractos de destino, llamados puertos de protocolo; cada uno de ellos identificado por medio de un número entero positivo. El sistema operativo local proporciona un mecanismo de interfaz que los procesos utilizan para especificar o accesar un puerto.

5 UDP y TCP

UDP
TCP

tente sin necesidad de realizar ningún cambio en los mecanismos de ruteo, encapsulando el contenido del paquete IPv6 en un paquete IPv4.

Existen diferentes formas de encapsulamiento: paquetes IPv6 encapsulados en paquetes IPv4 (Protocolo 41, 6to4, ISATAP, Tunnel Brokers), paquetes IPv6 encapsulados en paquetes GRE (Protocolo GRE) y paquetes IPv6 encapsulados en paquetes UDP (TEREDO).

Traducción

Las técnicas de traducción permiten un enrutamiento transparente en la comunicación entre los nodos de una red IPv6 y los nodos de una red IPv4. Pueden actuar de diferentes maneras y en zonas distintas: traduciendo encabezados IPv4 en encabezados IPv6 y viceversa, convirtiendo direcciones, convirtiendo APIs de programación o actuando en el intercambio de tráfico TCP o UDP.

UDP

Hemos descripto una Internet capaz de transferir datagramas IP entre computadoras arbitrarias, donde cada datagrama se rutea a través de la red, basándose en la dirección IP de destino. Ahora ampliamos el grupo de protocolos TCP/IP al agregar un mecanismo que distingue entre muchos destinos dentro de un host, permitiendo que varios programas de aplicación que se ejecutan en una computadora envíen y reciban datagramas en forma independiente.

Puertos

Los sistemas operativos permiten que varios programas de aplicación se ejecuten al mismo tiempo. Nos referimos a cada programa en ejecución como un proceso o tarea. Puede parecer natural decir que un proceso es el destino final de un mensaje. Sin embargo, especificar que un proceso en particular en una máquina en particular es el destino final para un datagrama es un poco confuso. Primero, como los procesos se crean y se destruyen de manera dinámica, los transmisores rara vez saben lo suficiente para identificar un proceso en otra máquina. Segundo, sería bueno poder reemplazar los procesos que reciben datagramas, sin tener que informar a los transmisores (ej. reinicio). Tercero, necesitamos identificar los destinos de las funciones implementadas sin conocer el proceso que implementa la función. También es importante saber, cuando un proceso maneja varias funciones, cuál debe decidir el proceso exactamente la función deseada por el transmisor.

En vez de pensar en un proceso como destino final, imaginemos que cada máquina contiene un grupo de puntos abstractos de destino, llamados puertos de protocolo; cada uno de ellos identificado por medio de un número entero positivo. El sistema operativo local proporciona un mecanismo de interfaz que los procesos utilizan para especificar o accesar un puerto.

La mayor parte de los sistemas operativos proporciona un acceso sincrónico a los puertos, es decir que los computos de un proceso se detienen durante una operación de acceso a puerto (por ej. si intentas acceder a un puerto antes de que llegan los datos, el SO bloquea el proceso y cuando llegan los datos se los pasa y lo reinicia). En general los puertos tienen memoria interna, para que los datos que llegan antes de que el proceso esté listo para aceptarlos no se pierdan. Para esto, el software de protocolo coloca los paquetes que llegan de un puerto de protocolo en particular en una cola de espera finita hasta que el proceso los extienda.

Protocolo de datagrama de usuario UDP

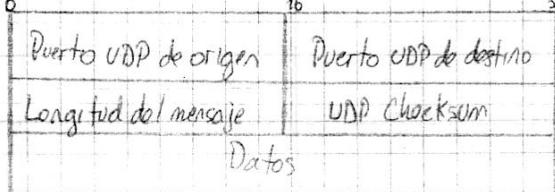
En el grupo de protocolos TCP/IP el protocolo de datagrama de usuario (UDP) proporciona el mecanismo primario que utilizan los programas de aplicación para enviar datagramas. El UDP proporciona puertos de protocolo utilizados para distinguir entre muchos programas que se ejecutan en la misma máquina. Esto quiere decir que, además de los datos, cada mensaje UDP contiene tanto el número de puerto de destino como el número de puerto de origen, haciendo posible que el software UDP en el destino entregue el mensaje al receptor correcto y que éste envíe una respuesta.

El UDP utiliza el IP subyacente para transferir un mensaje de una máquina a otra y proporciona la misma semantics de entrega de datagramas, sin conexión y no confiable, que IP. No emplea acuse de recibo para asegurarse de que llegan mensajes, no ordena los mensajes entrantes, ni proporciona retroalimentación para controlar la velocidad a la que fluye la información entre las máquinas. Por lo tanto, los mensajes UDP se pueden perder, duplicar o llegar sin orden. Además, los paquetes pueden llegar más rápido de lo que el receptor los puede procesar.

Un programa de aplicación que utiliza UDP acepta toda la responsabilidad por el manejo de problemas de confiabilidad, incluyendo la pérdida, duplicación y retraso de los mensajes, la entrega fuera de orden y la pérdida de conectividad. Muchos programas de aplicación que confían en el UDP trabajan bien en un ambiente local pero fallen dramáticamente cuando se utilizan en Internet.

Formato de los mensajes UDP

Cada mensaje UDP se conoce como datagrama de usuario, que conceptualmente consiste en dos partes: un encabezado UDP y un área de datos UDP.

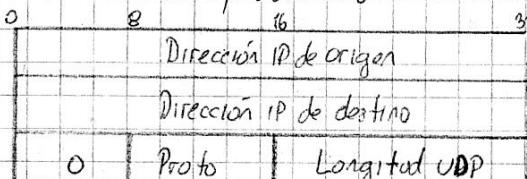


- Los campos Puerto de Origen (opcional) y Puerto de Destino contienen los números de puerto del protocolo UDP utilizados para el demultiplexado de datagramas entre los procesos que los esperan recibir.
- Longitud: contiene un conteo de los octetos en el datagrama UDP, incluyendo el encabezado y los datos (mín. 8).
- Checksum: es opcional, pero se recomienda ya que IP no verifica la porción de datos.

Pseudo-encabezado

El checksum UDP abarca más información de la que está presente en el datagrama UDP por sí solo. Para computarlo, el UDP añade un pseudo-encabezado al datagrama UDP, adjunta un octeto de ceros para rellenar y computa el checksum sobre todo el conjunto. Los agregados no se transmiten con el datagrama UDP, ni se incluyen en su longitud. Para computar el checksum, el software almacena uno en el campo y luego acumula una suma de complemento de 16 bits de todo el conjunto.

El propósito de utilizar un pseudo-encabezado es verificar que el datagrama UDP llega a su destino correcto. Como el encabezado UDP sólo especifica el número de puerto de protocolo pero también se necesita la dirección IP de la máquina para verificar el destino, el checksum cubre tanto la dirección IP como el datagrama UDP. Esto luego es checqueado por el software UDP de la máquina de destino.



El pseudo-encabezado consiste en 12 octetos de datos. El campo Proto contiene el código del tipo de protocolo IP (17 para UDP) y el campo de longitud no incluye el pseudo-encabezado en el cálculo. Para revisar el checksum, el receptor debe extraer los campos necesarios del header IP, ensamblarlos en el formato de pseudo-encabezado y recomputar la suma.

Existen contradicciones aparentes entre las reglas de la estratificación por capas y el cálculo del checksum UDP, ya que UDP debe interactuar con IP para obtener las direcciones IP. Pero esto se da por razones prácticas, ya que es imposible identificar plenamente un programa de aplicación de destino sin especificar la máquina de destino y porque se quiere realizar, de manera eficaz, la transformación de direcciones utilizadas por UDP e IP.

Multiplexado y demultiplexado

El software UDP acepta datagramas UDP de muchos programas de aplicación y los pasa a IP para su transmisión, así como también acepta datagramas UDP en tránsito del IP y los transfiere al programa de aplicación apropiado.

Conceptualmente, todo el multiplexado y desmultiplexado entre el software TCP y los programas de aplicación ocurre a través del mecanismo de puerto. En la práctica, cada programa de aplicación debe negociar con el sistema operativo para obtener un puerto de protocolo y un número de puerto asociado antes de poder enviar un datagrama UDP. Una vez que se asigna el puerto, cualquier datagrama que envíe el programa a través de él tendrá el número de puerto en el campo Puerto de Origen.

Mientras procesa la entrada, el UDP acepta datagramas entrantes del software IP y los desmultiplexa basándose en el puerto de destino UDP.

La forma más fácil de pensar en un puerto UDP es en una cola de espera. Cuando un programa de aplicación negocia con el sistema operativo la utilización de cierto puerto, el sistema operativo crea una cola de espera interna que puede almacenar los mensajes que lleguen. Cuando el UDP recibe un datagrama verifica si el número de puerto de destino corresponde a uno de los puertos que están en uso. Si no es el caso, envía un mensaje de error ICMP puerto no accesible y descarta el datagrama. Si encuentra una correspondencia, el UDP pone en cola de espera el nuevo datagrama en el puerto en que lo pueda accesar un programa de aplicación (si el puerto está lleno, ocurre un error y UDP descarta el datagrama entrante).

Números de puerto reservados

El asunto de asignar los números de puerto de protocolo es importante ya que dos computadoras necesitan estar de acuerdo en los números de puerto antes de que puedan interesar. Existen dos enfoques fundamentales para la asignación de puertos. El primero se vale de una autoridad central; todos se ponen de acuerdo en permitir que ella asigne los números de puerto conforme se necesiten y que publique la lista de todas las asignaciones. Luego, todo el software se diseña de acuerdo con la lista. Este enfoque recibe el nombre de universal y las asignaciones de puerto especificadas por la autoridad se designan como bien conocidas.

El segundo enfoque emplea la transformación dinámica. En este enfoque, los puertos no se conocen de manera global sino que siempre que un tránsito necesita un puerto, el software de red le asigna uno. Para conocer la asignación actual de puerto en otra computadora, es necesario enviar una solicitud que lo pregunte. La máquina objetivo responde al proporcionar el número de puerto correcto a utilizar.

Los diseñadores del TCP/IP adoptaron un enfoque híbrido que reasigna algunos números de puerto, pero que deja muchos de ellos disponibles para los sitios locales o programas de aplicación. Los números

de puerto asignados comienzan con valores bajos y se extienden hacia arriba dejando disponibles valores de números enteros altos para la asignación dinámica. En lo posible, otros protocolos de transporte que ofrecen los mismos servicios utilizan los mismos números de puerto que UDP.

Algunos números asignados de puerto son: 7 (ECHO), 9 (Desconect.), 20 (FTP), 23 (Telnet), 25 (SMTP), 37 (Time), 53 (DNS), 69 (TFTP), 80 (HTTP), 194 (IRC), etc.

TCP

Hasta el momento hemos explorado el servicio de entrega de paquetes sin conexión y no confiable que forma la base para toda la comunicación en Internet, así como el protocolo IP que lo define. Ahora introduciremos el segundo servicio más importante y mejor conocido de nivel de red, la entrega de flujo confiable, así como el Protocolo de Control de Transmisión (TCP) que lo define. TCP es un protocolo independiente de propósitos generales que se puede adaptar para utilizarlo con otros sistemas de entrega, ya que asume muy poco sobre la red subjacente.

En el nivel más bajo, las redes de comunicación por computadora proporcionan una entrega de paquetes no confiable; los paquetes se pueden perder o destruir debido a errores de transmisión, fallas de hardware o sobre-carga en las redes. Las redes que rutean los paquetes dinámicamente pueden entregarlos en orden, con retraso o duplicados.

En el nivel más alto, los programas de aplicación a menudo necesitan enviar grandes volúmenes de datos de una computadora a otra. Utilizar un sistema de entrega sin conexión y no confiable para las transferencias de gran volumen se vuelve tedioso, molesto y requiere la implementación de detección y solución de errores en cada programa, cosa que es bastante difícil. Como consecuencia, una meta de la investigación de protocolos de red ha sido encontrar soluciones de propósito general para el problema de proporcionar una entrega de flujo confiable, lo que posibilita construir una sola instancia de software de protocolos de flujo que sea utilizada por todos los programas de aplicación.

Características de la entrega confiable

La interfaz entre los programas de aplicación y el servicio TCP/IP de entrega confiable se puede caracterizar por 5 funciones:

- Orientación de flujo: cuando dos programas de aplicación transfieren grandes volúmenes de datos, pensamos en los datos como un flujo de bits, divididos en bytes. El servicio de entrega de flujo en la máquina de destino pasa al receptor exactamente la misma secuencia de bytes que le pasa el transmisor en la máquina de origen.

- Conexión de circuito virtual: la transferencia de flujo es similar a realizar una llamada telefónica. Antes de poder emplear la transferencia, los programas de aplicación, el transmisor y el receptor interactúan con sus respectivos sistemas operativos, informándose de la necesidad de realizar una transferencia de flujo. Conceptualmente, una aplicación realiza una llamada que la otra debe aceptar. Los módulos de software de protocolo en ambos SO se comunican, verificando que la transferencia esté autorizada y que los extremos estén listos. Luego de que se establecen todos los detalles, se informa a los programas de aplicación que se estableció una conexión y que la transferencia puede comenzar. Durante la transferencia, el software de protocolo en las dos máquinas se sigue comunicando para verificar que los datos se reciban correctamente. Si la comunicación no se logra ambas máquinas detectarán la falla y la reportarán a los programas apropiados. Se utiliza el término circuito virtual ya que la confiabilidad proporcionada depende del servicio de entrega de flujo pese a que la conexión se ve como un circuito de hardware.

- Transferencia con memoria intermedia: los programas de aplicación envían un flujo de datos a través del circuito virtual pasando repetidamente bytes al software de protocolo. Cuando transfiere datos, cada aplicación utiliza piezas del tamaño que encuentre adecuado. En el extremo receptor, el software de protocolo entrega bytes del flujo de datos en el mismo orden que se enviaron al programa de aplicación receptor, tan pronto como se reciben y verifican. El software de protocolo puede dividir el flujo en paquetes, fuera de los paquetes que transfiere el programa de aplicación. Para hacer eficiente la transferencia y minimizar el tráfico de red, las implementaciones recolectan datos suficientes de un flujo para llenar un datagrama largo antes de transmitirlo.

Para aplicaciones en las que los datos se deben entregar aunque no se tiene una memoria intermedia, el servicio de flujo proporciona un mecanismo de empuje que se utiliza para forzar una transferencia. En el extremo transmisor, un empuje obliga al software de protocolo a transferir todos los datos generados sin tener que esperar a que se tiene una memoria intermedia. Cuando llega al extremo receptor, el empuje hace que el TCP ponga los datos a disposición de la aplicación sin demora. Aún con el empuje, el software de protocolo puede dividir el flujo en formas inesperadas.

- Flujo no estructurado: el servicio de flujo TCP/IP no está obligado a formar flujos estructurados de datos (no marca fronteras). Los programas de aplicación que utilizan el servicio de flujo deben entender el contenido del flujo y ponerse de acuerdo sobre su formato antes de iniciar una conexión.

- Conexión Full Duplex: las conexiones proporcionadas por el servicio de flujo TCP/IP permiten la transferencia concurrente.

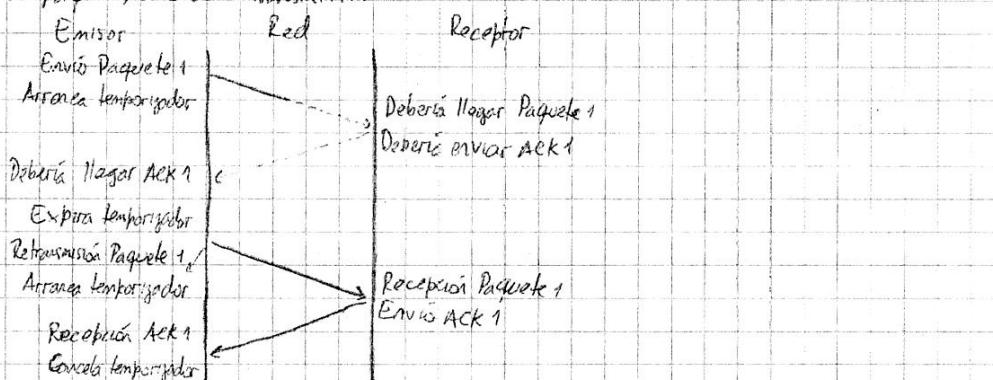
Mónte en ambas direcciones. Desde el punto de vista de una aplicación, una conexión full duplex consiste en dos flujos independientes que se mueven en direcciones opuestas, sin ninguna interacción abierta. El servicio de flujo permite que un proceso de aplicación termine el flujo en una dirección mientras los datos continúan moviéndose en la otra dirección (half-duplex). La ventaja de una conexión full duplex es que el software de protocolo subyacente puede enviar datagramas con información de control de flujo al origen, mientras lleva datos en la dirección opuesta. Este procedimiento reduce el tráfico en la red.

Comunicación confiable

Para proporcionar una transferencia confiable cuando el sistema subyacente de comunicación sólo ofrece una entrega no confiable de paquetes, se utiliza una técnica fundamental conocida como acuse de recibo positivo con retransmisión. La técnica requiere que un receptor se comunique con el origen y le envíe un mensaje de acuse de recibo (ACK) conforme recibe los datos. El transmisor guarda un registro de cada paquete que envía y espera un ACK antes de enviar el siguiente paquete. Además, arranca un temporizador y retransmite el paquete si éste expira antes de que llegue un ACK.



Cuando se pierde un paquete, se debe retransmitir.



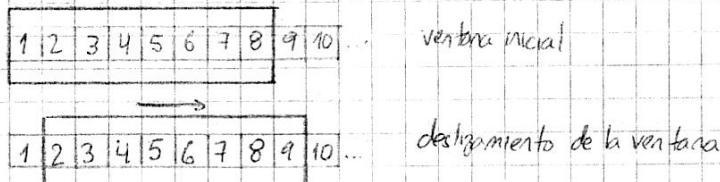
El problema final de confiabilidad surge cuando un sistema subyacente de entrega de paquetes los duplica (también puede ocurrir cuando hay largos retrasos y se retransmite prematuramente). La solución requiere acciones cuidadosas ya que tanto los paquetes como los acuses de recibo se pueden duplicar. Los protocolos confiables detectan los paquetes duplicados al asignar a cada uno un número de secuencia y obligar al receptor a recordar ese número de secuencia recibe. Para evitar la confusión causada por ACKs retrasados o duplicados, los protocolos de acuse

de recibos positivos envía los números de secuencia dentro de los ACKs, para que el receptor los pulse o reordenar correctamente con los paquetes.

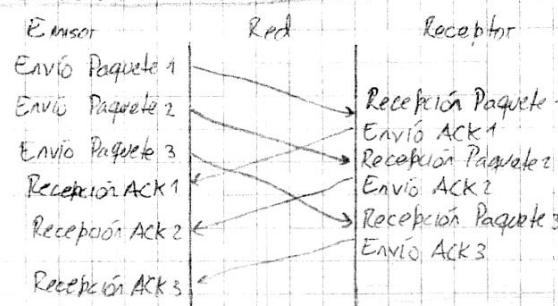
Ventana deslizable

El concepto de ventana deslizable hace que la transmisión de flujo sea eficiente. Para lograr confiabilidad, el transmisor envía un paquete y espera un ACK antes de enviar otro. Los datos solo fluyen en una dirección a la vez, incluso si la red tiene conexión full duplex. La red estará del todo ociosa durante el tiempo en que las máquinas retrasan sus respuestas. En conclusión, un protocolo simple de acuses de recibo positivos ocupa una cantidad sustancial de ancho de banda de red debido a que debe retrasar el envío de un nuevo paquete hasta que reciba un acuse de recibo del paquete anterior.

La técnica de ventana deslizable es una forma más compleja de acuse de recibo positivo y retransmisión. Los protocolos de ventana deslizable utilizan el ancho de banda de red de mejor forma, ya que permiten que el transmisor envíe varios paquetes sin esperar un acuse de recibo. El protocolo coloca una ventana pequeña y de tamaño fijo en la secuencia, y transmite todos los paquetes que residen dentro de la ventana.



Técnicamente, el número de paquetes sin acuse de recibo (transmitidos pero no recibió ACK) en un tiempo determinado depende del tamaño de la ventana y está limitado a un número pequeño y fijo. Una vez que el transmisor recibe un ACK para el primer paquete dentro de la ventana, miente la misma y envía el siguiente paquete. La ventana continuará moviéndose en tanto se reciben ACKs.



El desempeño de los protocolos de ventana deslizable depende del tamaño de la ventana y de la velocidad en que la red acepta paquetes. Con un tamaño de ventana de 1, un protocolo de ventana deslizable sería idéntico a un protocolo simple de acuse de recibo positivo. Al aumentar el tamaño de la ventana, es posible eliminar completamente

el tiempo ocioso de la red, es decir, en una situación estable, el transmisor puede enviar paquetes tan rápido como la red los pueda transferir. El punto principal es que, como un protocolo de ventana deslizable, en este modo maneja la red completamente autorizada de paquetes, con él se obtiene una generación de salida subsecuencialmente más alta que con un protocolo simple de acuerdo de recepción positivo.

Conceptualmente, en un protocolo de ventana deslizable siempre recordarás que los paquetes tienen acuerdo de recepción y mantiene un temporizador separado para cada paquete sin acuerdo de recepción. Si se pierde un paquete, el temporizador concluye y el transmisor reenvía el paquete. Cuando el emisor desliza su ventana, mueve hacia atrás todos los paquetes con acuerdo. En el extremo receptor, el software de protocolo mantiene una ventana análoga que acepta y acusa como recibidos los paquetes conforme llegan. Por lo tanto, la ventana divide la secuencia de paquetes en 3 partes: los paquetes a la izquierda de la ventana se transmitieron, recibieron y acusaron exitosamente; los paquetes a la derecha no se han transmitido; y los paquetes que quedan dentro de la ventana están en proceso de transmisión. El paquete con menor número en la ventana es el primer paquete en la secuencia para el que no se ha hecho un ACK.

Protocolo de control de transmisión

El servicio de flujo controlado proporcionado por el grupo de protocolos TCP/IP, el TCP, es un protocolo de comunicación, no una pieza de software.

TCP especifica el formato de datos y los avisos de recepción que intercambian dos computadoras para lograr una transferencia confiable, así como los procedimientos que la computadora utiliza para asegurarse de que los datos lleguen de manera correcta. También, especifica cómo el software TCP distingue el correcto entre muchos destinos en una misma máquina, y cómo las máquinas en comunicación resuelven errores como la pérdida o duplicación de paquetes. El protocolo también especifica cómo dos computadoras inicien una transferencia de flujo TCP, y cómo se ponen de acuerdo cuando se completa.

El protocolo no aclara los detalles de la interfaz entre un programa de aplicación y el TCP, es decir, no especifica los procedimientos exactos que los programas de aplicación invocan para acceder las operaciones del TCP. Esto es así para obtener flexibilidad y poder utilizar la especificación TCP en varios sistemas operativos.

Puertos, conexiones y puntos extremos

TCP permite que varios programas de aplicación en una máquina se comuniquen de manera concurrente y realiza el desmultiplexado del tráfico TCP entrante entre los programas de aplicación. Al igual que el UDP, TCP utiliza números de puerto de protocolo para identificar el destino final dentro de una máquina.

Cuando vemos los puertos UDP, pensamos cada puerto como una cola de salida en la que el software de protocolo coloca los datagramas entrantes. Los puertos TCP son mucho más complejos, ya que un número de puerto no corresponde a un solo objeto. De hecho, TCP se define según la abstracción de conexión, en la que los objetos que se van a identificar son conexiones de circuito virtual, no puertos individuales. El TCP utiliza la conexión, no el puerto de protocolo, como su abstracción fundamental; las conexiones se identifican por medio de un par de puntos extremos.

Puede ser natural asumir que un programa de aplicación sirve como el "punto extremo" de la conexión. Sin embargo, TCP define que un punto extremo es un par de enteros (host, puerto), en donde host es la dirección IP de un anfitrión y puerto es un puerto TCP en dicho anfitrión.

Una conexión estará identificada por los pares (host, puerto) que representan cada extremo. La abstracción de conexión permite que varias conexiones comparten un punto extremo sin que haya ambigüedad debido a que TCP asocia los mensajes entrantes con una conexión, y no con un puerto de protocolo (como ejemplo se tiene el caso del correo electrónico, que permite recibir correo de varios computadoras en forma concurrente en planteando sólo un puerto TCP local).

Como TCP es un protocolo orientado a la conexión, requiere que ambos puntos extremos estén de acuerdo en participar, es decir, antes de que el tráfico TCP pueda pasar a través de Internet, los programas de aplicación en ambos extremos de la conexión deben estar de acuerdo en que desean dicha conexión. Para hacerlo, el programa de aplicación en un extremo realiza una función de apertura pasiva, al contactar a su sistema operativo e indicar que aceptará una conexión entrante. En ese momento, el sistema operativo asigna un número de puerto TCP a su extremo de la conexión. El programa de aplicación en el otro extremo debe contactar a su sistema operativo mediante una solicitud de apertura activa para establecer una conexión. Los dos módulos de software TCP se comunican para establecer y llevar a cabo la conexión. Luego de creada, los procesos pueden comenzar a transmitir datos; los módulos de software TCP en cada extremo intercambian mensajes que garantizan la entrega confiable.

Segmentos, flujo y números de secuencia

TCP visualiza el flujo de datos como una secuencia de bytes que divide en segmentos para su transmisión. Por lo general, cada segmento viaja a través de Internet como un solo datagrama IP.

TCP utiliza un mecanismo especializado de ventana deslizante para solucionar dos problemas: unificar los

la transmisión eficiente y el control de flujo. Al igual que el protocolo de ventana deslizable, el mecanismo de ventana del TCP hace posible enviar varios segmentos antes de que llegue un aviso de recibo, aumentando la generación total de salida al mantener ocupada la red. La versión TCP del protocolo de ventana deslizable también soluciona el problema de control de flujo de extremo a extremo, al permitir que el receptor restrinja la transmisión hasta que tenga espacio suficiente en memoria intermedia para incorporar más datos.

El mecanismo TCP de ventana deslizable opera a nivel de byte, no de segmento ni de paquete. Los bytes del flujo de datos se numeran de manera secuencial y el transmisor guarda 3 punteros asociados con cada conexión, que definen una ventana deslizable. El primero marca el extremo izquierdo de la ventana, separando los bytes que ya se enviaron y fueron avisados de los bytes todavía no avisados. Un segundo puntero marca el extremo derecho de la ventana deslizable y define el byte más alto en la secuencia que se puede enviar antes de recibir más ACKs. El tercer puntero señala la frontera dentro de la ventana que separa los bytes ya enviados de los no enviados. El software de protocolo envía sin retraso todos los bytes dentro de la ventana por lo que, en general, la frontera dentro de la ventana se mueve rápidamente de izquierdo a derecho.

Describimos cómo se desliza la ventana TCP del transmisor y mencionamos que el receptor debe tener una ventana similar para ensamblar de nuevo el flujo. Pero, como las conexiones TCP son full duplex, se llevan a cabo dos transferencias al mismo tiempo en cada conexión, una en cada dirección. Por lo tanto, el software TCP mantiene dos ventanas en cada extremo, una que se desliza a lo largo del flujo de datos que se envía y otra que se desliza a lo largo de los datos que se reciben.

Ventana variable y control de flujo

En el protocolo TCP se permite que el tamaño de la ventana varíe. Cada aviso de recibo, que informa cuántos bytes se recibieron, contiene un aviso de ventana, que especifica cuántos bytes adicionales está preparado para aceptar el receptor. Podría pensarse en el aviso de ventana como la especificación del tamaño actual de la memoria intermedia del receptor. En respuesta a un aumento en el aviso de ventana, el transmisor incrementa el tamaño de su ventana deslizable y rápidamente responde a una disminución. El tamaño de la ventana cambia en el momento que se move hacia adelante.

La ventaja de utilizar una ventana de tamaño variable es que está proporcionada control de flujo así como una transferencia confiable. Si la memoria intermedia del receptor se llena, no puede aceptar más paquetes, así que envía un aviso de ventana más pequeño. En caso extremo, el receptor anuncia un tamaño de ventana 0 para detener todo la transmisión (salvo puntero enrgante activado o pruebas periódicas para resetear cronometro tridi). Cuando

Una memoria intermedia disponible, el receptor anuncia un tamaño de ventana distinto a 0 para activar nuevamente el flujo de datos.

TCP utiliza su esquema de ventana deslizable para resolver el problema de control de flujo extremo a extremo, pero no cuenta con un mecanismo explícito para el control de congestión (sobrecarga en las máquinas intermedias).

Según la implementación, el problema puede ser solucionado o empeorado.

Formato del segmento TCP

La unidad de transferencia entre el software TCP de dos máquinas se conoce como segmento. Los segmentos se intercambian para establecer conexiones, transferir datos, enviar avisos de recibido, anunciar los tamaños de ventana y finalizar conexiones.

Bit	0	4	10	16	24	31
Puerto fuente					Duerto destino	
Número de secuencia						
Número de aviso de recibir						
HLEN	Reserved	Code bits		Ventana		
Suma de verificación				Pointer de urgencia		
Opciones (si las hay)				Relayer		
Datos						

- Los campos Source Port y Destination Port contienen los números de puerto TCP que identifican a los programas de aplicación en los extremos de la conexión.
 - Sequence Number: identifica la posición de los datos del segmento en el flujo de datos del transmisor.
 - ACK Number: identifica el número de bytes que la fuente espera recibir después.
 - HLEN: especifica la longitud del encabezado de segmento medida en múltiplos de 32 bits.
 - Code Bits: con este campo se determina el propósito y contenido del segmento. Contiene:
 - URG: el campo de Puntero de Urgencia es válido.
 - ACK: el campo de cause de rechazo es válido.
 - PSH: el segmento solicita una operación push.
 - RST: reseteo de la conexión.
 - SYN: sincronizar números de secuencia.
 - FIN: el enisor ha llegado al final de su flujo de octetos.

- Window: especifica el tamaño de la memoria en memoria que los datos están dispuestos a aceptar).
- Checksum: suma de verificación de errores y 16 bits utilizada para verificar la integridad de los datos y del header.
- Urgent Pointer: especifica la posición dentro del segmento en la que terminan los datos urgentes.
- Opciones: se utiliza para negociar (por ej. tamaño máximo).

Datos urgentes

Aunque TCP es un protocolo orientado al flujo, algunas veces es importante que el programa en un extremo de la conexión envíe datos fuera de banda, sin esperar que el programa en el otro extremo de la conexión termine de considerar los octetos que ya están en flujo (por ej. para abortar procesos en sesiones de acceso remoto).

Para incorporar la señalización fuera de banda, TCP permite que el transmisor especifique los datos como urgentes, dando a entender que se debe notificar su llegada al programa receptor tan pronto como sea posible, sin importar su posición en el flujo. El protocolo especifica que, cuando se encuentra con datos urgentes, el TCP receptor debe notificar al programa de aplicación asociado con la conexión que entre en "modulidad urgente". Después de assimilar todos los datos urgentes, el TCP indica al programa de aplicación que regrese a su operación normal.

Este procedimiento se realiza utilizando el bit de código URG y el campo Urgent Pointer.

Opción de tamaño máximo de segmento

No todos los segmentos que se envían a través de una conexión serán del mismo tamaño. Sin embargo, ambos extremos necesitan acordar el tamaño máximo de los segmentos que transferirán. Una de las opciones posibles para el ejemplo Options permite que el software TCP especifique el tamaño máximo de segmento (MSS) que está dispuesto a recibir.

Si los dos puntos extremos residen en la misma red física, el TCP, por lo general, computará un MSS de tal forma que los datagramas IP resultantes correspondan con la MTU de la red. Si no es el caso, pueden intentar descubrir la mínima MTU a lo largo del camino entre ellos o pueden escoger un MSS de 536 (tamaño por defecto de un datagrama IP, 5% menos el tamaño estandar de los headers IP y TCP).

Escoger un MSS apropiado puede ser difícil, ya que el desempeño puede ser bajo tanto por tamaños de segmentos muy grandes o muy pequeños. Cuando el tamaño de segmento es pequeño la utilización de la red permanece baja. Cuando el tamaño de segmento es muy grande, los datagramas IP resultan muy grandes y generalmente se requerirá fragmentación, disminuyendo la producción de salida.

En teoría, el tamaño óptimo de segmento S ocurre cuando los datagramas IP que llevan los segmentos son tan grandes como sea posible sin requerir fragmentación en ninguna parte a lo largo del camino entre la fuente y el destino. En la práctica, encontrar S es muy difícil, ya que las implementaciones de TCP no incluyen un mecanismo

• mismo, para hacerlo, los routers en Internet pueden cambiar las rutas en forma dinámica (cambiando los MTUs) y el tamaño óptimo depende de los encabezados de protocolo de nivel más bajo.

Cálculo del checksum TCP

Para computar la suma de verificación TCP sigue un procedimiento igual al de UDP, cobra un pseudo-encabezado en el segmento, rellena con 0s para obtener un múltiplo de 16 bits y calcula la suma de 16 bits sobre todo el resultado. El pseudo-encabezado contiene la dirección IP de la fuente y del destino, un campo que indica el protocolo (en TCP es 6) y un campo que especifica la longitud total del segmento TCP. El propósito del pseudo-encabezado es el mismo que en UDP: permitirle al receptor que verifique que el segmento llegó a su destino correcto.

Acuses de recibo y retransmisión

Como TCP envía los datos en segmentos de longitud variable, y debido a que los segmentos retransmitidos pueden incluir más datos que los originales, los acuses de recibo no podrán resultarse fácilmente a los datagramas o segmentos. En cambio, se remiten a una posición en el flujo, utilizando los números de secuencia de flujo. El receptor recibe bytes de datos de los segmentos entrantes y reconstruye una copia exacta del flujo que se envía. Como los segmentos se pueden perder o llegar en desorden, el receptor utiliza los números de secuencia para recomponer los segmentos. En cualquier momento, el receptor tendrá algunos bytes reconstruidos contiguamente desde el comienzo del flujo, pero puede tener piezas adicionales del flujo que hayan llegado en desorden. El receptor siempre avisa recurso del prefijo contiguo más largo del flujo que se recibió correctamente. Un acuse de recibo TCP especifica el número de secuencia del siguiente byte que el receptor espera recibir.

Al esquema TCP de acuse de recibo se lo llama acumulativo porque reporta cuánto se ha acumulado de flujo. La ventaja de este esquema es que los acuses de recibo son fáciles de generar y no son ambiguos, además de que los ACK perdidos no necesariamente forzarán la retransmisión. Una gran desventaja es que el emisor no obtiene información sobre todas las transmisiones exitosas, sólo información sobre una sola posición en el flujo que se recibió. Una muestra se da cuando se pierde el primer segmento de una serie y el resto llega bien. Los ACKs siguientes especifican la posición del primer segmento y el transmisor no sabe si el resto de los datos llegó correctamente. Si esto ocurre se debe elegir entre dos esquemas intercalmente mutuas: enviar todo la secuencia de nuevo, o enviar sólo el primer segmento y esperar el acuse de recibo para decir cómo sigue, volviendo a un protocolo simple de acuse de recibo positivo.

Tiempo límite y retransmisión

Una de las ideas más importantes y complejas del TCP es parte de la forma en que maneja los timeouts y la retransmisión. TCP asume que el destino envía avisos de recibo siempre que recibe con éxito nuevos bytes del flujo de datos. Cada vez que envía un segmento, TCP inicia un temporizador y espera un ACK. Si se termina el tiempo antes de que llegue, TCP asume que el segmento se perdió o corrompió, y lo retransmite.

Es imposible para TCP saber con anticipación qué tan rápido regresan los ACKs al origen debido a la variedad de redes en Internet y los cambios debidos al tráfico. El software TCP debe incorporar las amplias diferencias en el tiempo necesario para llegar a varios destinos, así como los cambios en el tiempo necesario para llegar a cierto destino conforme varía la carga de tráfico.

TCP maneja los retrasos variables en Internet al utilizar un algoritmo adaptable de retransmisión. En esencia, TCP monitorea el desempeño de cada conexión y deduce valores razonables para el timeout. Conforme cambia el desempeño de una conexión, TCP revisa su valor de timeout, adaptándose al cambio.

Para recolectar los datos para un algoritmo aceptable, TCP registra la hora en que se envía cada segmento y la hora en la que se recibe un ACK para los datos en el segmento. Luego se computa el tiempo transcurrido, conocido como (tiempo) ejemplo de viaje en redondo (RTS). Siempre que obtiene un nuevo RTS, TCP ajusta su noción de tiempo de viaje en redondo (RTT) promedio para la conexión. Una técnica para hacerlo es considerar un factor constante α , entre 0 y 1, y utilizar la fórmula $RTT = \alpha * Old_RTT + (1-\alpha) * New_RTT$. El valor de α determina con qué rapidez responde el promedio a los cambios en el retraso.

Cuando envía un paquete, TCP computa un valor de timeout como una función de la estimación actual para viaje redondo. Las implementaciones antiguas de TCP se valían de un factor constante $B > 1$ y la fórmula $timeout = B * RTT$. El valor recomendado para B era 2, pero se han producido mejoras técnicas para el ajuste del timeout.

Medición precisa de RTS

En teoría, la medición de una RTS es trivial; consiste en restar la hora a la que se envía el segmento de la hora a la que llega el aviso de recibo. Sin embargo, surgen complicaciones debido a que TCP utiliza un esquema de avisos de recibo acumulativos. En un datagrama retransmitido, como ambos segmentos llevan exactamente los mismos datos, el receptor no tiene forma de saber si un ACK corresponde al datagrama original o al retransmitido. Este fenómeno se conoce como ambigüedad de aviso de recibo y se dice que los ACKs son ambiguos.

Si TCP asume que los ACKs pertenecen a la transmisión original, el RTT probablemente crecerá resultando en timeouts mayores. En situaciones transitorias el tiempo volverá a ser mayor y así sucesivamente. Si en cambio,

asocia los ACKs con la transmisión más reciente y ocurre un repentino aumento en el retraso extremo a extremos el porque se será retransmitido x el tiempo el ACK original, la RTS será muy pequeña y disminuirá el RTT para los próximos segmentos.

La estimación del RTT se puede establecer en un valor T , que sea de tal manera que el tiempo correcto de viaje en redondo resulte ligeramente mayor que algunos múltiplos de T . Las implementaciones de TCP que asocian los ACKs con la retransmisión más reciente llegarán a un estado estable con el RTT ligeramente menor que la mitad del valor correcto! (TCP envía cada segmento 2 veces aunque no haya perdida)

Algoritmo de Karn KARIN

La solución al problema anterior es sencilla; TCP no debe actualizar la estimación de RTT para los segmentos retransmitidos. Esta idea, conocida como algoritmo de Karn, evita el problema de todos los ACKs ambiguos al ajustar la estimación de viaje redondo únicamente para ACKs no ambiguos, es decir, relacionados con segmentos que solo se transmitieron una vez.

Por supuesto, una implementación simplista del algoritmo de Karn, que solo ignore los tiempos para los segmentos retransmitidos, también puede conducir a fallas. Supongamos que TCP envía un segmento después de un aumento significativo en el retraso. El timeout existente será demasiado pequeño y forzará la retransmisión. Si TCP ignora los ACKs para los segmentos retransmitidos, nunca actualizará la estimación y el ciclo continuará.

Para resolver dichos fallos, el algoritmo de Karn necesita que el transmisor combine los timeouts de transmisión con una estrategia de anulación del temporizador (timer backoff). La técnica de backoff computa un timeout inicial por medio de una fórmula similar a la anterior. Pero si se termina el tiempo y se procede a una retransmisión, TCP aumenta el valor de timeout. La técnica más utilizada por las implementaciones para computar el backoff nuevo incluye un factor multiplicativo α , y ajustar el nuevo valor a $\text{new_timeout} = \alpha * \text{timeout}$ (por lo general $\alpha=2$). Otras implementaciones utilizan una tabla de factores multiplicativos, lo que permite el backoff arbitrario en cada paso.

El algoritmo de Karn combina la técnica de backoff con la técnica de estimación de RTT para solucionar el problema de no incrementar las estimaciones de viaje redondo: cuando se compute la estimación de viaje redondo, ignorar los ejemplos que corresponden a los segmentos retransmitidos, pero utilizar una estrategia de backoff, y mantener el valor de timeout de un paquete retransmitido para los paquetes subsecuentes, hasta que se obtenga un ejemplo válido.

Respuesta al congestionamiento

Parecería como si el software TCP se hubiera diseñado considerando la interacción entre los puntos extremos de una conexión y los retrasos en la comunicación entre ellos. Sin embargo, en la práctica TCP también debe responder al congestionamiento en Internet, causado por una sobre carga de datagramas en uno o más puntos de conmutación. Cuando ocurre un congestionamiento, los retrasos aumentan y los routers comienzan a colgar en colas de salida a los datagramas hasta poderlos rutear. En el peor de los casos, la memoria de un router se llena y éste comienza a descartar datagramas.

Los puntos extremos, por lo general no conocen los detalles sobre dónde ocurrió un congestionamiento o por qué. Para ellos, el congestionamiento tan sólo significa un mayor retraso. Como la mayor parte de los protocolos de transporte utiliza el timeout y la retransmisión, se responde a un aumento en el retraso retransmitiendo datagramas, lo que empeora el congestionamiento. Si no se revisa, el incremento en el tráfico producirá mayor retraso, conduciendo a mayor tráfico, y así sucesivamente hasta que la red no pueda utilizarse. La condición se conoce como colapso por congestionamiento.

Para evitar esto, TCP debe reducir la velocidad de transmisión cuando ocurre un congestionamiento. Los routers pueden utilizar técnicas como la solicitud de disminución ICMP para informar a los hosts de un congestionamiento, pero protocolos como TCP pueden ayudar a evitar el congestionamiento al reducir automáticamente la velocidad de transmisión siempre que ocurre un retraso.

TCP recomienda la utilización de dos técnicas: arranque lento y disminución multiplicativa. Para cada conexión, TCP debe recordar el tamaño de la ventana del receptor. Para controlar el congestionamiento, TCP mantiene un segundo límite, llamado ventana de congestionamiento. En cualquier momento, TCP actúa como si el tamaño de la ventana fuera el mínimo entre lo indicado por el receptor y la ventana de congestionamiento.

En un estado de conexión no congestionada, la ventana de congestionamiento es del mismo tamaño que la ventana del receptor. La reducción de la ventana de congestionamiento reduce el tráfico que TCP injectará a la conexión. TCP asume que la mayor parte de la pérdida de datagramas viene del congestionamiento y se vale de la estrategia de disminución multiplicativa: cuando se pierda un segmento, reducir a la mitad la ventana de congestionamiento (mínimo 1 segmento) y aumentar exponencialmente el temporizador para la retransmisión en los segmentos que permanecan en la ventana permitida.

Como TCP reduce a la mitad la ventana de congestionamiento por cada pérdida, si las pérdidas continúan la ventana disminuye exponencialmente, reduciendo el volumen de tráfico y la velocidad de retransmisión. En estos extre-

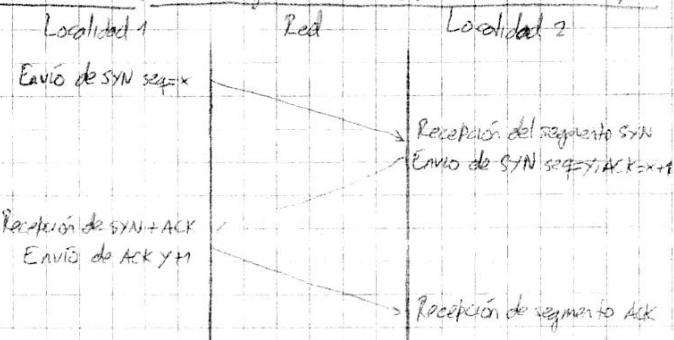
mos, TCP limita la transmisión a un solo datagrama y continua duplicando los valores de timeout antes de retransmitir. La idea es proporcionar una reducción rápida y significativa del tráfico, dandole tiempo a los routers para deshacerse de los datagramas en sus colas de espera.

Para recuperarse cuando termina el congestionamiento, TCP no duplica la ventana de congestionamiento, debido a que produciría un sistema instable que oscilaría entre poco tráfico y congestionamiento. En cambio, TCP implementa la técnica de arranque lento para aumentar la transmisión: siempre que se arranque el tráfico en una nueva conexión o se aumente el tráfico después de un período de congestionamiento, activar la ventana de congestionamiento con tamaño de 1 segmento y aumentarla 1 segmento cada vez que llegue un ACK. Los arranques lentos evitan saturar la Internet con tráfico adicional luego de que se libere un congestionamiento o cuando comienzan repentinamente nuevas conexiones.

Para evitar el aumento demasiado rápido del tamaño de la ventana y no causar congestionamiento adicional, TCP agrega una restricción más. Una vez que la ventana de congestionamiento llega a la mitad de su tamaño original (antes del congestionamiento), TCP entra en una fase de prevención de congestionamiento y hace más lenta la tasa de incremento. En esta fase, aumenta el tamaño de la ventana en 1 segmento sólo si se tienen avisos de recepción para todos los segmentos en la ventana.

Establecimiento de una conexión TCP

Para establecer una conexión, TCP utiliza un saludo (handshake) de 3 etapas.



El primer segmento del saludo se puede identificar porque tiene activo el bit SYN en el campo de código. El segundo mensaje tiene tanto el bit SYN como el bit ACK activos, indicando tanto el aviso de recepción del primer segmento como el hecho de que se continua con el intercambio. El mensaje final del saludo es sólo un aviso de recepción y nada más se utiliza para informar al destino que ambos extremos están de acuerdo en establecer una conexión.

Una conexión se puede establecer desde cualquier extremo o desde ambos al mismo tiempo, ya que el handshake está diseñado para cubrir esto. Una vez establecida la conexión, los datos pueden fluir en ambas direcciones por igual.

El saludo de tres etapas es necesario y suficiente para la sincronización correcta entre los dos extremos de la conexión. Esto se debe a que TCP se construye sobre un servicio de entrega no confiable de paquetes. Sucederían algunos problemas si las solicitudes originales y retransmitidas llegan mientras se establece la conexión o si las solicitudes retransmitidas se retrasan hasta que se establezca, utilice y termine una conexión. Un saludo de 3 etapas, más la regla de que TCP ignore solicitudes adicionales de conexión después de que se establezca la misma, resuelve estos problemas.

Números de secuencia inicial

El saludo de 3 etapas realiza dos funciones importantes: garantiza que ambos lados estén listos para transferir datos y que lo sepan y permita a ambos partes acordar un número de secuencia inicial. Los números de secuencia son enviados y reconocidos durante el saludo. Cada máquina debe seleccionar un número de secuencia inicial en forma aleatoria que se utilizará para identificar bytes en el flujo que se está enviando; los números de secuencia no pueden coincidir siempre con el mismo valor.

La máquina A, que inicia un saludo, transfiere un número de secuencia inicial \oplus en el campo de secuencia del primer segmento SYN. La máquina B recibe el SYN, registra el número de secuencia y responde enviando su número de secuencia inicial \ominus en el campo de secuencia así como un reconocimiento que especifica el byte $x+1$ esperado por B. En el mensaje final del saludo, A envía un ACK de la recepción del mensaje de B especificando el byte $x+1$.

Es posible enviar datos junto con los números de secuencia iniciales en los segmentos de saludo. En estos casos, el software TCP debe manejar los datos hasta que se complete el saludo. Una vez que se estableció la conexión, el software TCP puede liberar los datos y entregarlos rápidamente al programa de aplicación.

Terminación de una conexión TCP

Dos programas que utilizan TCP para comunicarse pueden terminar la conversación volviéndose de la operación doble de manera interna. TCP utiliza una modificación del saludo de 3 etapas para cerrar conexiones. Recordemos que las conexiones TCP son de tipo full duplex y contienen dos transferencias de flujo independientes, una en cada dirección. Cuando un programa de aplicación informa al TCP que ya no tiene más datos para enviar, este cerrará la conexión en una dirección. Para cerrar la mitad de una conexión, el envío TCP termina de transmitir los datos restantes, espera la recepción de un acuse de recibo y entonces envía un segmento con el bit FIN activado. El receptor TCP reconoce el segmento FIN e informa al programa de aplicación en su extremo que no tiene más datos disponibles.

Random Early Discard

Cuando un router en pausa descarta datagramas en tránsito porque su memoria intermedia está llena, puede causar un efecto interesante. En el caso de que los datagramas pertenezcan a una única conexión TCP, los perdidos causarán que ésta entre en arranque lento. Pero cuando los datagramas llevan segmentos de varias conexiones TCP, el descarte puede causar sincronización global. Es más probable que un router descarte 1 segmento de varias conexiones que varios segmentos de una conexión (debido a que los datagramas se multiplexan). Esta pérdida simultánea causa que todas las conexiones TCP entran en arranque lento al mismo tiempo.

Para evitar esto, se utiliza el esquema de Random Early Discard (RED). Un router que implementa RED usa dos variables frontera para monitorear posiciones en la cola: T_{min} y T_{max} . La operación general de RED se puede describir en 3 reglas:

- Si la cola contiene menos de T_{min} datagramas, agregar el datagrama en tránsito a la cola.
- Si la cola contiene más de T_{max} datagramas, descartar el datagrama en tránsito.
- Si la cola contiene entre T_{min} y T_{max} datagramas, descartar el datagrama en tránsito con probabilidad p .

La aleatoriedad de RED permite que en lugar de esperar que la cola se sobrecargue y provoque varios arranques lentos, un router lenta y aleatoriamente descarte paquetes a medida que aumenta la congestión.

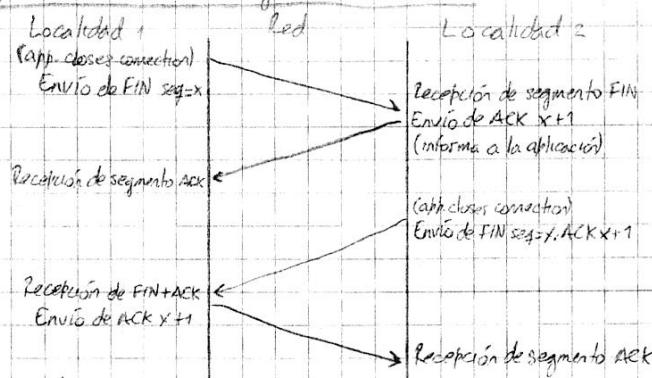
La clave para que funcione RED radica en la elección de las fronteras T_{min} y T_{max} , y la probabilidad de descartar la. T_{min} debe ser lo suficientemente grande para asegurar la utilización alta del enlace saliente. T_{max} debe ser mayor que T_{min} por un valor importante (por ej. el doble o más) para evitar las oscilaciones globales del descarte. El valor de p se calcula en cada datagrama dependiendo de la relación entre el tamaño actual de la cola y las fronteras; entre los valores T_{min} y T_{max} , la probabilidad p varía de 0 a 1 linealmente.

Este esquema lineal debe retocarse para evitar sobreexacciones. Esto se debe a que el tráfico en Internet funciona en ráfagas y lo que resulta en rápidas fluctuaciones en la cola de un router. Si RED vio en ese momento una ráfaga, los datagramas posteriores en una ráfaga tendrían altas probabilidades de ser descartados. Pero un router debería evitar descartar datagramas innecesariamente porque hacerlo repercutiría negativamente en el control TCP. Entonces, si una ráfaga es corta, no es recomendable descartar datagramas porque la cola no se sobrellenaría. Pero tampoco se puede ignorar el descarte inefectivamente porque ráfagas largas causarían sobreexceso.

La respuesta a este problema reside en una técnica "prestada" de TCP: en lugar de usar el tamaño actual de la cola en cualquier instante, RED computa un promedio ponderado de tamaño de cola y lo usa para determinar la probabilidad. El

promedio se actualiza con cada dato para entrar k, con la fórmula $\text{avg} = (1-\alpha) * \text{old_avg} + \alpha * \text{current_queve size}$, donde α es un valor entre 0 y 1.

Una vez que la conexión se cerró en una dirección dada, TCP rechaza más datos en esa dirección. Mientras tanto, los datos pueden continuar flujiendo en la dirección opuesta hasta que el emisor la cierra. Por supuesto, los ACKs continúan flujiendo hacia el emisor, incluso después de que la conexión se ha cerrado. Cuando ambas direcciones se cierran, el software TCP en cada punto extremo borra sus registros de la conexión.



La diferencia entre el saludo de 3 etapas empleado para establecer conexiones y el usado para interrumpirlas se presenta luego de que una máquina recibe el segmento FIN inicial. En vez de generar un segundo segmento FIN inmediatamente, TCP envía un ACK y luego informa a la aplicación de la solicitud de interrupción. Como informar al programa de aplicación y obtener una respuesta puede tomar un tiempo considerable, el ACK evita la retransmisión del segmento inicial FIN durante la espera. Por último, cuando el programa de aplicación instruye al TCP para que interrumpa la conexión completamente, éste envía el segundo segmento FIN y la localidad original responde con el ACK final.

Reseteo de una conexión TCP

Normalmente, un programa de aplicación se vale de la operación close para interrumpir una conexión cuando termina de utilizarla. Algunas veces se presentan condiciones anormales que obligan a un programa de aplicación o al software de red a interrumpir una conexión. TCP proporciona una funcionalidad de reseteo para estas excepciones anormales.

Para resetear una conexión, un lado inicia la interrupción enviando un segmento con el bit RST activado en el campo de control. El otro lado responde a un segmento de reseteo inmediatamente interrumpiendo la conexión. TCP también informa al programa de aplicación que ha ocurrido un reseteo. Un reseteo es una interrupción instantánea, lo que significa que la transferencia en ambas direcciones se interrumpe de manera inmediata y se liberan recursos como los buffers.

Entrega forzosa de datos

Hemos dicho que TCP es libre de dividir el flujo de datos en segmentos para su transmisión sin considerar el tamaño de transferencia que utiliza el programa de aplicación. La mayor ventaja de esto es la eficiencia que se obtiene.

Aún cuando el procesamiento en memoria intermedia mejora el desempeño de la red, puede interferir con algunas aplicaciones por ej. sesiones remotas que responden a cada pulsación de tecla. Además, el uso de memoria intermedia en el receptor obliga a hacer algo más que forzar al emisor a transmitir.

TCP proporciona la operación Push que un programa de aplicación puede utilizar para forzar la entrega de los datos en el flujo de transmisión sin esperar a que se los almacene en memoria intermedia. La operación push también solicita a TCP que active el bit PSH en el campo de código del segmento, así los datos se entregan al programa de aplicación en el extremo de recepción.

Números de puerto TCP reservados

Como UDP, TCP controla la asignación dinámica y estática de puertos mediante un conjunto de asignación de puertos bien conocidos para programas llamados con frecuencia, pero se deja la mayor parte de los números de puerto disponibles para que el sistema operativo los asigne conforme los programas los necesitan. Algunos números de puerto bien conocidos son: 2(eco), 11(active users), 17(quote), 21(FTP), 22(ssh), 25(SMTP), 37(time), 53(DNS), 80(www), 161(SNMP), etc.

Aun que los números de puerto TCP y UDP son independientes, los diseñadores decidieron utilizar el mismo número de puerto para cualquier servicio accesible desde UDP y TCP (ej. 53 DNS).

Síndrome de ventana fija

Los investigadores que participaron en el desarrollo de TCP observaron un serio problema de desempeño que puede presentarse cuando las aplicaciones del emisor y del receptor operan a velocidades diferentes. Para entenderlo, recordemos que TCP almacena en memoria intermedia los datos de entrada y consideremos que la aplicación del receptor elige leer los datos de entrada un byte a la vez. Cuando se establece una conexión, el receptor TCP asigna un buffer de K bytes y utiliza el campo WINDOW en los segmentos de ACK para anunciar el tamaño disponible de bytes al emisor. Si la aplicación del emisor genera datos rápidamente, el emisor TCP transmitirá segmentos con datos para todo la ventana. Eventualmente, el emisor recibirá un ACK que especifique que toda la ventana está llena y que no queda espacio adicional en el buffer del receptor.

Cuando la aplicación del receptor lee un byte de datos del buffer lleno queda disponible un espacio de 1byte. Hemos dicho que cuando queda espacio disponible en el buffer, TCP genera un ACK que utiliza el campo WINDOW para informar al emisor. En este caso, el receptor anunciará una ventana de 1byte. Cuando haya consumptione del espacio disponible, el emisor TCP

respondrá con la transmisión de un segmento que contenga 1 byte de datos.

Aún cuando el anuncio de la ventana de 1 solo byte trate de manera correcta conservando toda la memoria intermedia del receptor, el resultado es una serie de pequeños segmentos de datos. El emisor TCP debe comprender un segmento que contenga un byte de datos, colocar el segmento en un datagrama y transmitir el resultado. Cuando la aplicación del receptor lee otro byte, TCP generará otro ACK, lo cual ocasionará que el emisor trasmite otro segmento que contenga 1 byte de datos. La interacción resultante puede llegar a establecerse en un estado en el cual TCP envíe un segmento separado para cada byte de datos.

La transferencia de segmentos pequeños ocupa ancho de banda de la red innecesariamente e introduce una sobrecarga computacional. Lo primero se debe a que cada datagrama lleva solo 1 byte de datos; la relación creciendo-datos es grande. La sobrecarga computacional se origina debido a que TCP debe procesar cada segmento tanto en el emisor como en el receptor. El software TCP del emisor debe asignar espacio en el buffer, formar un encabezado de segmento y calcular la checksum. Asimismo, el software IP del emisor debe encapsular el segmento en un datagrama, calcular la checksum del encabezador, rotar el datagrama y transferirlo hacia la interfaz de red apropiada. En la máquina receptora el IP debe verificar la checksum del encabezado y transferir el segmento hacia el TCP. Es éste que debe verificar la checksum del segmento, examinar el número de secuencia, extraer el dato y colocarlo en una memoria intermedia.

Un emisor también puede ocasionar que cada segmento contenga una pequeña cantidad de datos. Esto podría suceder con una implementación TCP que genera que envíe datos cada vez que estén disponibles, y una aplicación emisora que genere 1 byte de datos por vez. También podría pasar que una aplicación genere datos en bloques de tamaño fijo B, y el emisor TCP extraiga datos de la memoria intermedia en bloques de tamaño M. Si $M < B$, el último bloque en la memoria intermedia puede ser pequeño.

Este problema se conoce como sistema de ventana fija (fixed window) y consiste en paquetes de red enviando una pequeña cantidad de espacio disponible y cada segmento transportando una pequeña cantidad de datos.

Prevención de SWS

Ahora, las especificaciones de TCP incluyen la heurística necesaria para prevenir el SWS. La heurística utilizada en una máquina emisora evita la transmisión de cantidades pequeñas de datos en cada segmento. Otra heurística establecida en la máquina receptora evita la emisión de incrementos pequeños al los anuncios de ventana que fuerzan activar paquetes pequeños de datos. Tener heurísticas en ambos extremos ayuda en casos que alcance tal e

en la correcta implementación de los procedimientos. Ambos deben tener características para emisor y receptor.

Prevención en el lado del receptor

La heurística que utiliza un receptor para evitar las ventanas faltas es fácil de entender. En general, un receptor mantiene un registro interno de la ventana disponible en el momento, pero retrasa los anuncios de incremento de la ventana al emisor hasta que la ventana avance una cantidad significativa (depende del tamaño de la memoria intermedia del receptor y el MSS). TCP define la cantidad como el mínimo entre la mitad de la memoria intermedia del receptor y el número de bytes de datos en un segmento de máximo tamaño.

El procedimiento para evitar las ventanas faltas en el lado del receptor evita el anuncio de ventanas pequeñas en caso de que una aplicación del receptor extraiga bytes de datos lentamente. Así, el emisor siempre recibirá incrementos extensos en la ventana actual, permitiendo la transferencia de segmentos grandes.

En resumen, antes de enviar el anuncio de una ventana actualizada, tiene de anunciar una ventana igual a 0, esperar hasta que se obtenga un espacio disponible que sea equivalente a por lo menos la mitad del tamaño total de la memoria intermedia o igual a un segmento de tamaño máximo.

RESUMEN

Acoses retardados

Se han tomado dos enfoques para implementar la prevención de las ventanas faltas en el lado del receptor. Uno consiste en acusar cada segmento recibido, pero no anunciar el incremento en la ventana hasta alcanzar el límite especificado en la heurística. El otro método consiste en retrasar el envío de un ACK cuando la prevención de ventanas faltas especifica que la ventana no es suficientemente grande como para anunciarse. Los estándares recomiendan retrasar los acuses de recibo.

El retraso de los ACKs tiene ventajas y desventajas. La mayor ventaja reside en que el retraso de los ACKs puede reducir el tráfico y, por lo tanto, mejorar el desempeño (por ejemplo, si llegan datos adicionales en el período de retraso, un solo ACK reconocerá todos los datos). Si la aplicación del receptor genera una respuesta inmediata, el retraso puede permitir que el ACK sea incorporado a un segmento de datos. En casos en que la aplicación receptora lee los datos tan pronto como llegan, un pequeño retraso permite a TCP enviar un solo segmento de acuse de recibo y que anuncie la actualización de ventanas sin retraso (se enviaría un segmento para cada uno).

La principal desventaja del retraso en los ACKs es que si se retrasa mucho tiempo el acuse, el emisor TCP retransmitirá el segmento. Las retransmisiones innecesarias reducen el desempeño debido a que desperdician ancho de banda de la red, además de causar sobrecargas computacionales en ambos extremos. Además, como TCP usa los ACKs para estimar los RTT, retrasarlos puede alterar la estimación y alterar los tiempos de retransmisión.

Para evitar problemas botánicos, TCP establece un límite para el retraso de un ACK. Las implementaciones no pueden retrasar un ACK por más de 500 ms. Además, para garantizar que TCP reciba un número suficiente de confirmaciones de los segmentos, se recomienda enviar al menos uno de cada dos segmentos.

Prevención en el lado del emisor

La heurística utilizada por un emisor TCP para evitar las ventanas fotorras es sorprendente y elegante. Recordemos que el objetivo es evitar el envío de segmentos pequeños y que una aplicación emisora tiene que enviar datos en bloques relativamente pequeños. Para lograr este objetivo, el TCP emisor debe permitir a la aplicación hacer múltiples llamadas para escribir y debe reunir los datos transferidos en cada llamada antes de transmitirlos en un solo segmento grande. Es decir, un emisor TCP debe retrasar el envío de un segmento hasta que pueda acumular una cantidad razonable de datos. Esta técnica se conoce como clumping.

Surge la cuestión de cuánto tiene que esperar TCP antes de transmitir datos. Por un lado, si TCP es demasiado lento, la aplicación tendrá retrasos demasiados largos. Algo importante es que TCP no puede saber si tiene que esperar pues no puede saber si la aplicación generará más datos en un futuro cercano. Por otro lado, si TCP no espera lo suficiente, los segmentos serán pequeños y el desempeño disminuirá.

Los protocolos prensos a TCP introducen retrasos antes de transmitir cada pulsación de tecla para determinar si el usuario continúa pulsando teclas. Pero como TCP es tan diseñado para ser general, puede usarse para diversas aplicaciones. Los caracteres pueden viajar a través de una conexión TCP ya sea porque el usuario tipea o porque el programa está transfiriendo un archivo. Un retraso fijo no es óptimo para todas las aplicaciones.

La técnica que utiliza un emisor TCP para evitar el envío de paquetes pequeños es flexible; el retraso depende del desempeño actual de la Internet. La prevención de la ventana fotorra en el lado del emisor se llama self clocking ya que no computa retrasos. En cambio TCP usa la llegada de un ACK para disparar la transmisión de paquetes adicionales.

SIAME

La heurística se resume de esta forma: cuando una aplicación emisora genera datos adicionales para enviarse sobre una conexión para la cual se transmitieron datos previos, todavía no acuñados, colocar los datos nuevos en la memoria intermedia de salida, pero no enviar segmentos adicionales hasta que haya suficientes datos para enviar un segmento de tamaño máximo. Si todavía se está a la espera cuando llega un ACK, enviar todos los datos que se hayan acumulado en la memoria intermedia. Aplicar la regla incluso cuando el usuario solicita una operación push.

Si una aplicación genera en byte de datos por vez, TCP enviará el primer byte inmediatamente. Sin embargo, hasta que llegue el ACK, TCP acumulará bytes adicionales en la memoria intermedia. Entonces, si la aplicación es rápida en comparación con la red, los segmentos sucesivos contendrán muchos bytes cada uno. Si la aplicación es lenta, se enviarán segmentos pequeños sin retrasos largos.

Conocido como algoritmo de Nagle, esta técnica es especialmente elegante pues requiere poca carga computacional. Un arbitrio no necesita tener temporizadores separados para cada conexión ni examinar un reloj cuando una aplicación genera datos. Aunque esta técnica se adapta a combinaciones arbitrarias de retraso en la red, MSS y velocidad de aplicación no baja el desempeño en casos convencionales.

Para entender por qué, observemos que las aplicaciones optimizadas, para alto desempeño no generan datos en byte a la vez (hacerlo induciría sobrecargas innecesarias en el SO). En cambio, esas aplicaciones escriben grandes bloques de datos en cada llamada. Así, la memoria intermedia de salida de TCP tiene suficientes datos para al menos un segmento de tamaño máximo. Además, como la aplicación produce datos más rápido de lo que TCP los puede transferir, la memoria intermedia del envío permanece casi llena, y TCP no retrasa la transmisión. Como resultado, TCP continúa enviando segmentos en la medida que Internet lo puede tolerar, mientras la aplicación mantiene la memoria intermedia llena.

DNS

La Internet funciona asignando una dirección, local o global, IP óptica a cada punto extremo (host, servidor, router, interfaz, etc.). Pero, sin la posibilidad de asignar un nombre correspondiente a cada recurso, cada vez que se quiera acceder a un recurso disponible en la red (por ej. el sitio www.example.com) sería necesario saber su dirección IP física (como 192.168.34.160). Con la gran cantidad de hosts y sitios Web que hay, esto sería imposible.

Para resolver este problema, se creó el concepto de name servers para permitir que las propiedades o atributos de un recurso con nombre mantendrán en una ubicación bien conocida; la idea básica es que la gente recuerda fácilmente el nombre de algo (en especial si es descriptivo) en vez de una dirección numérica.

Cuando hay un name server presente en la red un host sólo necesita saber la dirección física de un name server, y el nombre del recurso que quiere acceder. Usando esta información, puede encontrar la dirección (u otra propiedad atributo guardado) del recurso interrogando (querying) el name server. Los recursos pueden ser agregados, reubicados, cambiados o borrados en una sola ubicación: el name server, y la información nueva estará inmediatamente disponible para todo host que use ese name server. Un name server es simplemente una base de datos especializada que traduce nombres a propiedades (en general, direcciones IP) y viceversa. Los name servers simplifican el manejo de una red y las hace más eficientes y receptivas a los cambios.

6 DNS y Firewall

Si una aplicación genera un byte de datos por vez, TCP enviará el primer byte inmediatamente. Sin embargo, hasta que llegue el ACK, TCP acumulará bytes adicionales en la memoria intermedia. Entonces, si la aplicación es rápida en comparación con la red, los segmentos sucesivos contendrán muchos bytes cada uno. Si la aplicación es lenta, se enviarán segmentos pequeños sin retrasos largos.

Conocido como algoritmo de Nagle, esta técnica es especialmente elegante pues requiere poca carga computacional. Un arbitrio no necesita tener temporizadores separados para cada conexión ni examinar un reloj cuando una aplicación genera datos. Aunque esta técnica se adapta a combinaciones arbitrarias de retraso en la red, MSS y velocidad de aplicación no baja el desempeño en casos convencionales.

Para entender por qué, observemos que las aplicaciones optimizadas, para alto desempeño no generan datos en byte a la vez (hacerlo induciría sobrecargas innecesarias en el SO). En cambio, esas aplicaciones escriben grandes bloques de datos en cada llamada. Así, la memoria intermedia de salida de TCP tiene suficientes datos para al menos un segmento de tamaño mixto. Además, como la aplicación produce datos más rápido de lo que TCP los puede transferir, la memoria intermedia del emisor permanece casi llena, y TCP no retrasa la transmisión. Como resultado, TCP continúa enviando segmentos en la medida que internet lo permite, mientras la aplicación mantiene la memoria intermedia llena.

DNS

La Internet funciona asignando una dirección, local o global, IP única a cada punto extremo (host, servidor, router, interlo, etc.). Pero, sin la posibilidad de asignar un nombre correspondiente a cada recurso, cada vez que se quiera acceder a un recurso disponible en la red (por ej. el sitio www.example.com) sería necesario saber su dirección IP física (como 192.168.34.166). Con la gran cantidad de hosts y sitios Web que hay, esto sería imposible.

Para resolver este problema, se creó el concepto de name servers para permitir ciertas propiedades o atributos de un recurso con nombre mantenidas en una ubicación bien conocida; la idea básica es que la gente recuerda fácilmente el nombre de algo (en especial si es descriptivo) en vez de una dirección numérica.

Cuando hay un name server presente en la red un host sólo necesita saber la dirección física de un name server y el nombre del recurso que quiere acceder. Usando esta información, puede encontrar la dirección (u otra propiedad atributo guardado) del recurso interrogando (querying) el name server. Los recursos pueden ser agregados, reubicados, cambiados o borrados en una sola ubicación: el name server, y la información nueva estará inmediatamente disponible para todo host que use ese name server. Un name server es simplemente una base de datos especializada que traduce nombres a propiedades (en general, direcciones IP) y viceversa. Los name servers simplifican el manejo de una red y las hace más eficientes y receptivas a los cambios.

Si un name server no está disponible, entonces el host no podrá acceder a ningún recurso en la red. De esta manera el name server es un recurso crítico; es mejor tener más de un name server en caso de fallos.

La solución inicial a este problema fue introducir name servers Primario y Secundario (incluso se ven más). Si el name server Primario no respondía a una query, el host reintentaría usando el name server Secundario. Pero agrupar todos las transacciones en el name server Primario disminuiría el rendimiento global; además cuando este esté inoperable, la transacción tendría que esperar un poco antes de reintentar con el Secundario, y así sucesivamente. La mayoría del software de name server usa alguna forma de algoritmo de round-robin en la lista de name servers para intentar distribuir carga y acortar los tiempos de respuesta.

A medida que una red crece, el número de nombres en el name server aumenta mucho. Esto causa 3 problemas:

- Organización: encontrar una entrada en la base de datos de nombres se vuelve más lento a medida que se recorren millones de entradas buscando el deseado. Se necesita organizar o indexar los nombres.
- Escalabilidad: si cada host accede los name servers de la red, la carga se vuelve muy alta. Se necesita distribuir la carga a través de varios name servers.
- Administración: con muchos registros de nombres en la base de datos, la administración se vuelve cada vez más difícil, ya que muchos administradores intentan actualizar registros al mismo tiempo. Se necesita un método para separar (delegar) la administración de estos registros de nombre (recurso).

Estas necesidades desembocaron en la creación y evolución del Sistema de Nombres de Dominio de Internet (DNS).

Sistema de Nombres de Dominio

El DNS de Internet es solo una implementación específica del concepto de name server optimizado para las condiciones de Internet.

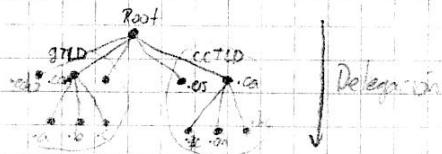
Dominios y Delegación

DNS usa una estructura de nombres en forma de árbol (jerárquica). En la parte superior del árbol está la raíz, seguida por los Dominios de Nivel Superior (TLDs), después los Dominios de Segundo Nivel (SLDs), y luego cualquier número de niveles interiores, todo uno separado por un punto.

Los TLDs se dividen en dos tipos:

- TLDs genéricos (gTLD): .com, .edu, .net, .org, .mil, etc. Existe una subcategoría de esparsos ods (sTLDs), con registro limitado como .area, .museum, .travel, .jobs, etc.

- TLDs de Código de País (ccTLD): esca una secuencia estandar de dos letras definida por la ISO (.es., .uk., .ar., .ca., etc.).



Lo que comúnmente se llama nombre de dominio (ejemplo: .com) es en realidad una combinación de un nOMBRE SLD y un nOMBRE TLD y se escribe de izquierda a derecha con el nivel más bajo en la jerarquía a la izquierda y el nivel más alto a la derecha: sld.tld.

El término SLD es técnicamente preciso ya que define todos en el segundo nivel de la jerarquía de nombres de dominio, pero puede ser confuso, por ejemplo cuando se usan ccTLDs (por ejemplo .com, .uk). Por convención, el término nombre de dominio describe una entidad delegada como example.com.

Autoridad de dominios

Los conceptos de autoridad y delegación son parte clave de la jerarquía de DNS. Cada nodo dentro de la jerarquía de nombres de dominio es asignado a una autoridad, una organización o persona responsable por el manejo del nodo. Tal organización o persona se dice que maneja el nodo autoritativamente. La autoridad de un nodo en particular puede a su vez delegar la autoridad para niveles más bajos de ese nodo dentro de la jerarquía de nombres de dominio.

La autoridad de la raíz yace con la ICANN, que asumió esta responsabilidad en 1998. Parte de los objetivos de ICANN al establecerse fue clarificar la posición de la jerarquía bajo su responsabilidad a la competencia comercial. Para facilitar esto, se creó el concepto de registrante acreditado, organizaciones a las que la ICANN delega responsabilidades limitadas para la venta y administración de partes de la jerarquía de nombres de dominio.

Los gTLDs son autoritativamente administrados por la ICANN y delegados a una serie de registrantes acreditados.
Los ccTLDs son delegados por la ICANN a los países individuales para propósitos administrativos. Cada autoridad tiene,
a su vez, delegar a niveles más bajos en la jerarquía, es decir, puede delegar a cualquier otra parte que es autoritativa.
Cada nivel de la jerarquía puede delegar el control autoritativo al siguiente nivel.

En los ccTLDs, cada país define sus propias reglas para la delegación. Algunos países deciden administrar a nivel nacional y delegar a cada provincia usando un código (ejemplo. .nx, .us). Otros países optan por una segmentación funcional en sus modelos de delegación (co:comunidad, ac:académico, etc) (ejemplo. .co.uk).

La delegación dentro de un dominio puede no tener límites y es decidida por la autoridad delegada. Por ej. en EEUU se definen ciudades dentro de dominios de estados; city.ny.us; sería la ciudad city, en el estado de Nueva York en EEUU y company.city.ny.us; sería el nombre de dominio de company en esa ubicación. Leer un nombre de dominio

de derecha a izquierda rastreará la delegación.

www.example.com

www.example.com está constituido por www y example.com. La parte de nombre de dominio example.com fue delegada de un registrante gTLD, que a su vez fue delegado de ICANN. La parte www fue delegada por el dueño del dominio ya que es autoridad delegada para el nombre example.com. El dueño posee todo a la izquierda del nombre de dominio delegado example.com.

La parte más hacia la izquierda (www en este caso) se llama host name. Solo por convención, los sitios web usan el host name www. La parte de host puede referirse a un host name real o a un nombre de servicio.

Toda computadora que esté conectada a Internet y sea accesada usando un name server tiene un host name. Por ejemplo: www.example.com (servicio web), ftp.example.com (servidor de transferencia de archivos), pc17.example.com (computadora normal), accounting.example.com (sistema de cuentas). Un host name debe ser único dentro del nombre de dominio delegado, pero puede ser cualquier cosa que el dueño quiera.

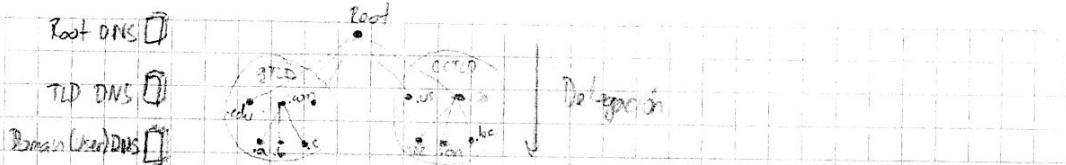
www.us.example.com

Deducimos que el nombre de dominio es example.com y www probablemente indica un sitio web, lo que dejó la parte us. Esta parte fue asignada por el dueño de example.com (que es autoritativo) y se llama subdominio. En este caso la autoridad delegada para example.com decidió que su organización es mejor servida por estructura de subdominio basada en el país. Podría haber delegado la responsabilidad internamente al subdominio de EEUU para la administración de este subdominio, lo que a su vez podría haber creado una estructura basada en plantas. www.cleveland.us.example.com podría indicar el sitio web de la planta de Cleveland en la organización de EEUU de example.com.

En resumen, el dueño puede delegar, de cualquier forma que quiera, todo lo que está a la izquierda del nombre de dominio que posezca que le sea delegado. El domo delegado es también responsable de administrar esta delegación, o sea, asumir la tarea de controlar DNS conteniendo información autoritativa (o registros) para su nombre de dominio. La unidad de delegación es llamada zona en la documentación.

Organización y estructura de DNS

DNS de Internet funciona con la misma estructura de delegación de nombres de dominio que acabamos de describir. Hay name servers (servidores que tienen software DNS) en cada nivel de la jerarquía de delegación y la responsabilidad de control el name server hace con el control autoritativo en ese nivel.

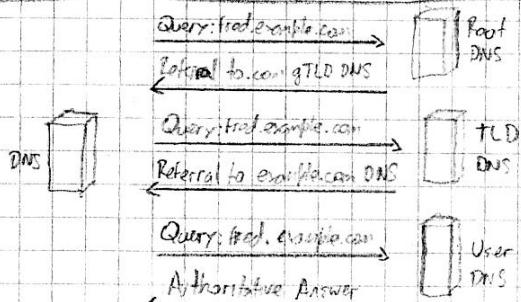


Los servidores raíz (root-servers) son los recursos más críticos en Internet. Son responsabilidad de la ICANN pero son operados por un consorcio bajo un acuerdo de delegación. Hay 13 root-servers en el mundo. Los root-servers son conocidos por todo servidor DNS en el mundo y son el punto de partida de toda query; se usa un archivo especial de zona que se distribuye con todo el software DNS. Para mayor robustez, hay copias de cada root-server distribuidas por el mundo y todas comparten la misma dirección IP (se utiliza anycast para enviar datos).

Los servidores TLD son operados por varias organizaciones y agencias (bajo varios acuerdos con ICANN) llamadas Operadores de Registro.

El dueño de un nombre de dominio tiene la autoridad y, por lo tanto, la responsabilidad de administrar los servidores de usuario (o nombre de dominio) (debe haber mínimo 2 para tener robustez). En muchos casos, el dominio delega la responsabilidad a un ISP, una compañía de web hosting o a un registrante. Otra opción es manejar sus propios name servers y delegar la autoridad y responsabilidad para los servidores DNS de los subdominios a otras partes de la organización.

Cuando un name server no puede resolver una query para un nombre de dominio, la query se pasa a un root-server, que refiere el servidor TLD apropiado, que a su vez refiere un servidor de usuario apropiado.



Componentes del sistema DNS

Un DNS incluye 3 partes: datos que describen los dominios, uno o más programas DNS y un programa o librería de resolución.

Un solo servidor DNS puede soportar varios dominios. Los datos para cada dominio describen propiedades globales del dominio y sus hosts o servicios. Estos datos se definen en forma de Registros de Recurso (RR) organizados en archivos de zonas.

Un programa DNS típicamente hace 3 cosas: leer un archivo de configuración que define las zonas para las que es responsable (dependiendo de la funcionalidad del programa) leer un archivo de conf. que describe comportamientos (como cachear o no) y responder a queries de hosts locales o remotos.

Un programa o librería de resolución está localizado en cada host y provee los medios para traducir una solicitud del usuario en operaciones para servidores DNS usando protocolos de transporte.

Zonas y archivos de zona

Los name servers soportan lo que se llama zona. Una zona es descripta usando un archivo de zona que traduce el nombre de dominio en entidades operacionales (hosts, mail servers, servicios, etc.). Los subdominios delegados por el dueño del nombre de dominio también pueden ser descriptos usando archivos de zona separados. Por lo tanto, un archivo de zona describe, usando RRs textuales, la parte del nombre de dominio que está siendo manejada por el software DNS; una zona designa una porción operacional y autoritativa del nombre de dominio administrada por un name server. Un archivo de zona generalmente consiste de:

- Datos que describen las propiedades de la zona, conocidos como start of Authority RR (obligatorio).
- Datos autoritativos para todos los hosts dentro de la zona, generalmente usando A RRs.
- Datos que describen información global para la zona, como registros MX describiendo los mail servers del dominio y registros NS describiendo los name servers que son autoritativos para el dominio.
- En el caso de delegación de subdominios, los name servers responsables por ellos, usando registros NS.
- En el caso de delegación de subdominios, uno o más glue records que permiten al name server acceder a los name servers de los subdominios, usando registros A.

Formato de archivos de zona

Un archivo de zona contiene 3 tipos de entrada:

- Comentarios: todo comentario comienza con ; y continúa hasta el fin de la línea.
- Directivas: toda directiva comienza con ; y se utilizan para controlar el procesamiento de los archivos de zona.
- Registros de recursos: se usan para definir las características/propiedades o entidades contenidas en el dominio.

Ej: \$TTL 12 h ;directiva

\$ORIGIN example.com.

@ IN SOA ns1.example.com hostmaster.example.com. (

2014020400 "Serial number"

3h refresco

15m update after

3w expires

2100m repeat after TLL

)

IN NS ns1.example.com.

Contenido de un archivo de zona

En general, un archivo de zona contendrá los siguientes RR y directivas:

- Directiva \$TTL: define el tiempo de vida por defecto para la zona o dominio, que es el tiempo que un RR puede ser cacheteado por otro servidor DNS (obligatoria).
- Directiva \$ORIGIN: el nombre de dominio para la zona que se está definiendo (opcional).
- Start of Authority RR: debe aparecer como el primer RR en un archivo de zona. Describe las características globales de la zona o dominio (obligatorio, sólo uno).
- Name Server RR: definen name servers que son autoritativos para la zona o dominio. Debe haber al menos dos NS RR, en un archivo de zona. Pueden referenciar name servers en este dominio o en uno externo (obligatorio).
- Mail Exchanger RR: define los mail servers para la zona. Si el dominio no provee servicio de e-mail, no hay necesidad de MX RR's. Pueden referenciar mail servers en este dominio o en uno externo (opcional).
- Address RR: definen las direcciones IPv4 de todos los host o servicios que existen en la zona, y se requiere que sean visibles públicamente. Los entradas IPv6 se definen usando AAAA RR's (opcional).
- CNAME RR: define un alias RR, que permite que un host o servicio sea definido como el alias de otro host (opcional).

Directiva \$TTL

Todo RR puede tener un tiempo de vida opcional especificado en segundos. La directiva \$TTL define el tiempo de vida por defecto que se aplica a cada RR que no lo haya definido explícitamente. En el contexto DNS, el tiempo de vida define el tiempo en segundos que un registro puede ser cacheteado por otro name server o un resolviendo. La sintaxis es \$TTL time-in-seconds (o a 2147483647) (recomendado > 1 día).

El TTL determina dos características operacionales de DNS: la carga de acceso (con TTL bajo se fuerzan queries DNS más frecuentes, lo que eleva la carga operacional en el name server de la zona) y la propagación de cambios (el valor de TTL representa el tiempo máximo que cualquier cambio tardaría en propagarse desde el name server de la zona a todos los usuarios).

Directiva \$ORIGIN

Esta directiva define el nombre de dominio que será adjuntado a cada nombre incompleto (o no calificado) en una RR; se adjunta un valor a los nombres que no terminan con un punto. La sintaxis es \$ORIGIN domain-name. Se pueden tener varias directivas \$ORIGIN en un archivo y se aplican desde el punto en que se definen en adelante. Esta directiva no es obligatoria.

Registro de recurso SOA

Este registro define las características y atributos clave para la zona o dominio. Es el registro más complejo e importante de un archivo de zona. Su sintaxis es: name TTL class RR name-server e-mail mrefresh rretry expiry NX.

- name: el símbolo @ se usa para notar el valor actual de \$ORIGIN que generalmente es el dominio del dominio.
- TTL: si no se explicita aquí se usa el definido en \$TTL.
- class: define la clase de registro; generalmente se usa IN (Internet), que es la clase por defecto.
- RR: el tipo de registro (SOA en este caso).
- name-server: define el Maestro Primario para esta zona (name server autoritativo). Puede ser interno o externo al dominio.
- e-mail: define un e-mail administrativo para la zona, al que se enviará correo reportando errores o problemas. Se recomienda usar la casilla reservada hostmaster pero cualquier correo válido sirve.
- serial number: define el número de serie asociado actualmente con la zona. Debe cambiarse cada vez que se hace un cambio en el dominio. Este valor se utiliza durante transferencias de zona para determinar si ha cambiado el archivo de zona. La convención es usar un formato yyymmddss.
- rrefresh: cuando se alcanza este valor, el name server esclavo de esta zona tratará de leer el SOA RR del maestro de la zona (transferencia de zona). Según este valor se determina cuán rápidos se propagan los cambios de maestro a esclavo.
- rretry: define el intervalo de reintentos si el esclavo falla en establecer contacto con el maestro en un refresh.
- expiry: define el tiempo después del cual los registros de la zona se asumen como ya no autoritativos (no se responden más queries para la zona). Cuando el esclavo logra contactar al maestro de la zona, este contador se reinicia. Si expiry es alcanzado, el esclavo dejará de responder queries y se dice que la zona está muerta.
- NX: define el tiempo que una respuesta negativa es cacheada. Mientras NX valga, una query fallida devolverá Name Errors (NXDOMAIN). Al expirar, se reintentará la operación.

Registro de recurso NS

Este registro define los name servers autoritativos (al menos 2) para el dominio o zona. La sintaxis de un NS RR es: name TTL class RR target-name.

- name: puede ser un nombre de dominio calificado o no calificado, @ o un espacio en blanco (dato previo de nombre).
- RR: en este caso es NS.

- target-name: define un nombre server autoritativo para el dominio (si está en el dominio debe haber un registro A correspondiente).

Registro de recurso MX

Este registro define los mail servers de un dominio o zona. La sintaxis es: name ttl class rr preference name

- rr: en este caso es MX.

- preference: indica la prioridad del mail server; mientras más bajo el número, mayor la prioridad.

- name: define un mail server para el dominio (si está dentro del dominio debe haber un A RR correspondiente).

Registro de recurso A

Este registro define la dirección IPv4 de un host particular en el dominio o zona. La sintaxis de un A RR es:

name ttl class rr ipv4

- name: el nombre del host.

- rr: en este caso es A.

- ipv4: define la dirección IPv4 física para el host. Todo host que se desea hacer públicamente visible también se define usando registros A.

Se puede tener varios hosts para una dirección y varias direcciones para un host.

Registro de recurso CNAME

Este registro define un alias para un host existente (definido por un registro A). La sintaxis de un CNAME RR es:

name ttl class rr canonical-name

- name: denota el alias.

- rr: en este caso es CNAME.

- canonical-name: el nombre real al que se manda el alias. Se recomienda no encastrar CNAMEs.

Usar CNAME causa que el nome server trabaje más ya que tanto el CNAME como el RR aliasado deben ser buscados. En name servers grandes este trabajo extra puede ser un problema. Se recomienda sólo usarlos cuando el host único está ubicado en un dominio externo.

Delegación de subdominios

Esta técnica es utilizada para delegar completamente la responsabilidad por un subdominio a otro nome server (también parecida usa un sólo archivo de zona para las estructuras del subdominio, llamada subdominio virtual). Supongamos que queremos una estructura con example.com nombre de dominio y us.example con nombre de subdominio.

Para disminuir la carga administrativa en la zona, esta técnica asume que la responsabilidad por el subdominio

será delegada completamente al administrador de zona de `us.example.com`, quien será responsable por los archivos de zona del subdominio y sus name servers. En lo que concierne a los servidores TLD y las autoridades de registro de Internet, los subdominios no existen. Toda query para algo que termine con `example.com` será referida a los name servers de la zona `example.com`. De la misma manera, esos name servers son responsables de referir la query a los name servers del subdominio (éstos sólo los ven los name servers del dominio, son invisibles para los gTLD servers).

Ej: \$TTL 2d, archivo de zona de `example.com`

\$ORIGIN example.com.

② IN SOA ns1 hostmaster [

)

IN NS ns1.us.example.com.

IN NS ns2.us.example.com.

IN MX mail.example.com.

NS1 IN A 199.168.0.3

NS2 IN A 199.168.0.4

MAIL IN A 199.168.0.5

\$ORIGIN us.example.com.

IN NS ns3.us.example.com.

IN NS ns4.us.example.com.; servicios auxiliares

NS3 IN A 10.10.0.24; glue record

El registro A para `ns3` es lo que se llama un glue record. Los glue records son necesarios para permitir que una query DNS para el subdominio devuelva una referencia conteniendo tanto el nombre del name server como su dirección IP. Si este registro no estuviera, una query tratando de resolver `us.example.com` tendría que resolver primero `ns3.us.example.com`. Pero como `ns3` está bajo el dominio `us.example.com`, resolver `ns3.us.example.com` requiere resolver `us.example.com`, causando una dependencia circular, que se rompe con el glue record al permitir que el name server de `example.com` provea la dirección IP de `ns3.us.example.com` directamente al que la pide.

En la práctica, los servidores TLD proveen la dirección IP cada SLD name server dentro del dominio o no, para minimizar el número de operaciones en una query. Cuando se realiza una query a un gTLD, este name server provee los glue records para los name servers de todos los SLDs. Estos glue records fueron definidos y capturados al registrarse el dominio.

En los casos que se delegan subdominios, el name server del dominio toma el papel de TLD name server, y debe proveer las direcciones IP de los name servers de los subdominios como respuesta a queries para un subdo-

```

$TTL 2d; archivo de zona de us.example.com
$ORIGIN us.example.com
@      IN SOA ns1.hostmaster(
        )
        IN NC NS3.US.example.com.
        IN NC NS2.US.example.com.
        IN MX mail.US.example.com.

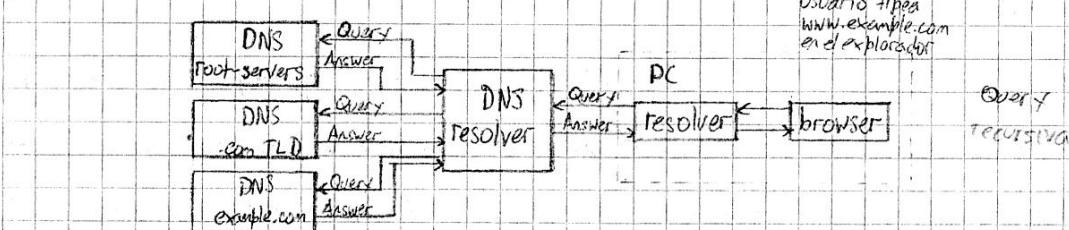
NS3   IN A 10.10.0.24
NS1   IN A 192.168.0.3; "glue" record
; otros servicios

```

El registro A para ns1.example.com es un glue record pero no es estrictamente necesario porque ya debe estar disponible en un resolvelor por alguna query previa. Para hacer una query para el subdominio us.example.com, el dominio example.com debe haber sido querido primero. Como ns1.example.com es uno de los name servers para example.com su dirección IP ya es conocida por el resolvelor que maneja la query del subdominio.

Queries DNS

La mayor tarea que lleva a cabo un name server autoritativo es responder queries de un resolvelor DNS local o remoto, o de otro name server actuando como resolvelor. Un resolvelor PC es la librería de software instalada en cada PC, que se usa para tratar la petición de usuario o aplicación en una query DNS. Una query típica podría ser "cuál es la dirección IP de www.example.com".



Un name server o resolvelor puede tener archivos de zona que lo definen como autoritativo para algunos dominios y esclavo para otros; y puede ser configurado para proveer cacheo, direccional queries u otros comportamientos.

La mayoría de las queries que recibe un DNS server serán para dominios para los que no tiene conocimiento, es decir, para los que no tiene archivos de zona locales. El software DNS permite al name server responder de formas distintas a este tipo de query. Hay 3 tipos de query: recursiva, iterativa e inversa.

Queries recursivas

Una query recursiva es una en la que el name server o resolvelor responde completamente (o devuelve un error). Los name servers no están obligados a soportar queries recursivas, y el resolvelor negocia su uso mediante algunos bits [REDACTED] en los encabezados de la query. Hay 3 posibles respuestas para una query recursiva:

- La respuesta a la query acompañada por cualquier registro CNAME que pueda ser útil. La respuesta indicará si los datos son autoritativos o cacheados.
- Un error indicando que el host o dominio no existe (NXDOMAIN). Esta respuesta también puede contener registros CNAME que apuntaran al host no existente.
- Una indicación de error temporal (como no poder acceder otro name server por errores en la red).

En una query recursiva, un name server trasteará, en nombre del cliente, el camino de name servers autoritativos a través de la red para obtener la respuesta real a la pregunta. El camino de una query recursiva para encontrar la dirección IP de www.example.com hecha a un name server que no es autoritativo para example.com podría ser así:

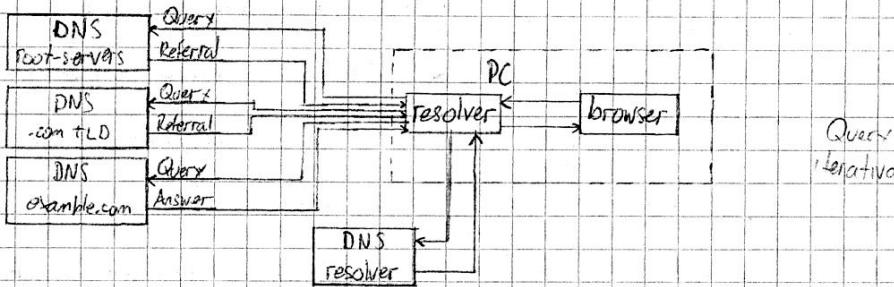
El usuario tipa www.example.com en un explorador. Este pide a su resovedor local la dirección IP de www.example.com. El resovedor DNS busca el nombre en sus tablas locales (su cache), pero no lo encuentra, entonces envía una query a un root-server por la dirección. El root-server sólo soporta queries iterativas, y responde con una lista de name servers que son autoritativos para el gTLD .com (esta se llama referencia). El resovedor DNS elige uno de los servidores gTLD autoritativos y le envía una query para la dirección IP de www.example.com. El name server gTLD sólo soporta queries iterativas, y responde con los name servers autoritativos para example.com. El resovedor DNS elige uno de ellos y le envía una query para la dirección IP de www.example.com. Supongamos que el archivo de zona para example.com define a www.example.com como un registro CNAME para joe.example.com; el name server autoritativo responde con el registro CNAME www.example.com y el registro A para joe.example.com (supongamos que es 192.168.154.2). El resovedor DNS envía la respuesta joe.example.com = 192.168.154.2 (junto con el registro CNAME www=joe) al resovedor local del cliente original, éste envía www.example.com = 192.168.154.2 al explorador del usuario. Finalmente, el explorador envía un solicitud a 192.168.154.2 para la página web.

Para decidir qué name server se usa cuando hay varios para elegir, se utilizan algoritmos que distribuyen la carga y aseguran el resultado más rápido posible (por ej. BIND mide tiempos de respuesta y elige el menor).

Queries Iterativas

Una query iterativa es una donde el name server puede proveer una respuesta parcial a la query y lo devolver en error. Los name servers deben soportar queries iterativas. Hay 4 respuestas posibles para una query iterativa:

- La respuesta a la query acompañada por cualquier registro CNAME que haya sido usado al definir el nombre. La respuesta indicará si los datos son autoritativos o cacheador.
- Un error indicando que el dominio o host no existe (NXDOMAIN). Esta respuesta puede contener registros CNAME que hagan apuntar al host no existente.
- Una indicación de error temporal.
- Una referencia; una lista de 2 o más name servers y direcciones IP cercanos al nombre de dominio requerido. Pueden o no ser los name servers autoritativos para el dominio final en la query. Una referencia es la respuesta normal de un root o TLD servers, ya que sólo soportan queries iterativas.



El camino de una query iterativa para encontrar la dirección IP de www.example.com hecha a un name server que soporta queries iterativas pero no es autoritativo para example.com podría ser de esta manera:

El usuario tipa www.example.com en un explorador. Este envía una solicitud para la dirección IP de www.example.com a su resoldedor. El resoldedor en un host manda una query iterativa para la dirección IP de www.example.com a su resoldedor DNS local. El resoldedor DNS busca www.example.com en sus tablas locales, pero no lo encuentra, por lo que responde con una referencia conteniendo la lista de root-servers. El resoldedor elige uno de los root-servers y le envía una query para la dirección IP de www.example.com. El root-server responde con una lista de name servers autoritativos para el TLD genérico .com. El resoldedor elige uno de los gTLD servers autoritativos y lo envía una query para la dirección IP de www.example.com directamente a ese name server. Este responde al resoldedor con los name servers autoritativos para el SLD example.com. El resoldedor elige uno de los SLD name servers autoritativos y le envía una query para la IP de www.example.com. El name server autoritativo responde con el registro CNAME www y el registro A para www.example.com, 192.168.254.2 (mismas subconsultas que para recursive queries). El resoldedor envía www.example.com = 192.168.254.2 al explorador. Finalmente, el explorador envía una solicitud a 192.168.254.2 para la página web.

Técnicamente, un resoldedor debe ser capaz de seguir referencias, pero los resoldedores instalados en la mayoría de los sistemas no lo hacen. En general, los resoldedores DNS usados por PCs o estaciones de trabajo deben soportar

tar queries recursivas para evitar devolver referencias al resolutor local.

Queries inversas

Una query inversa mapea un RR a un dominio(s): cuál es el nombre de dominio para este registro MX. El soporte para queries inversas estuvo siempre definido como opcional en las especificaciones de DNS y estaba permitido no implementarlo, algo que casi todos los name servers hicieron. Por lo tanto, las queries inversas no eran muy utilizadas y se volvieron obsoletas.

Podría parecer que las queries inversas se utilizan para encontrar un nombre de host dada una dirección IP, pero esto no es así. Este proceso se llama reverse mapping y se vale de queries recursivas e iterativas con el dominio especial IN-ADDR.ARPA.

Reverse mapping

Dado un nombre de dominio, una query DNS normal intentaría determinar su dirección IP. Sin embargo, hay ocasiones en las que es útil ser capaz de determinar el nombre de un host dada una dirección IP particular. Aunque a veces esto se requiere para diagnósticos, se utiliza frecuentemente con motivos de seguridad para rastrear un hacker o un spamer. Muchos sistemas de correo usan reverse mapping para proveer autenticación utilizando políticas DNS de los que queda para confirmar que la dirección IP específica representa el host indicado.

Para realizar reverse mapping usando queries iterativas y recursivas, los discendentes de DNS definen un nombre de dominio especial llamado IN-ADDR.ARPA.

Dominio IN-ADDR.ARPA

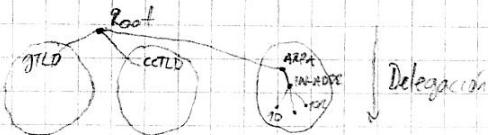
La estructura de un nombre de dominio normal es jerárquica, empezando por la raíz. Un nombre de dominio se escribe de izquierda a derecha, pero la estructura jerárquica se ve de derecha a izquierda. Tomando www.example.com, el nodo más arriba en la jerarquía DNS es la raíz, definida por el punto silencioso. Luego está el TLD .com, seguido por .example (SLD); y finalmente, el más bajo es www, que es el nombre de host. Para permitir que una dirección IPv4 sea usada en una operación de query, debe ser convertida en un nombre de dominio.

Tomenos la dirección IPv4 192.168.254.17, que define el host 17 en el rango de direcciones clase C 192.168.254.x. En este caso, la parte más importante (el nodo más alto) está a la izquierda (192), no a la derecha. Es lo es un poco molesto y haría imposible construir una estructura de árbol sensible que facilite la búsqueda.

La solución es invertir el orden de la dirección y construir la jerarquía bajo el dominio especial IN-ADDR.ARPA.

ARPA es TLD e IN-ADDR es SLN).

La última parte de la dirección IPv4(17) es la dirección de host y los hosts se definen dentro de un archivo de zona, por lo que lo ignoraremos. El resultado es 254.168.192.IN-ADDR.ARPA.



Se construye un archivo de zona que describe todos los hosts en la zona reverse mapped, usando registros PTR.

Ej: \$TTL 2d

\$ORIGIN 254.168.192.IN-ADDR.ARPA.

@ IN SOA ns1.example.com. hostmaster.example.com. (

)

IN NS ns1.example.com.
IN NS ns2.example.net.

2 IN PTR ns1.example.com.
4 IN PTR mail.example.com.
16 IN PTR joe.example.com.
17 IN PTR www.example.com.

Registro de recurso PTR

Este registro mapea una dirección IPv4 a un host particular en el dominio o zona. La sintaxis es:

ej: name ttl class rr name

-Name: la dirección del host generalmente.

-rr: en este caso PTR.

-name: define el nombre que una query por la dirección devolverá.

Una dirección IPv4 puede ser mapeada a varios hosts.

Mantenimiento de Zona

Para simplificar la operación de múltiples name servers, es útil que una sola fuente pueda actualizar varios servidores. Este proceso llamado mantenimiento de zona, puede involucrar transferencias de archivos de zona desde un servidor DNS hacia otro (entre un maestro DNS y un esclavo DNS para la zona).

El tiempo que se tarda en transmitir cambios en archivos de zonas es determinante para la velocidad en que los cambios en información de la zona se propagan a través de Internet. El diseño inicial de DNS permitía propagar cambios usando full zone transfer (AXFR), pero Internet era mucho más simple en esa época. El deseo de acelerar el proceso de propagación de actualizaciones de zona, minimizando recursos, resultó en varios cambios en este aspecto del diseño e implementación DNS, desde incremental zone transfer (IXFR) y mensajes NOTIFY hasta el concepto de actuali-

zonas dinámicas (DDNS).

Full zone transfer

Las especificaciones originales de DNS contemplaron que los name servers esclavos (secundarios) para la zona someterían al name server maestro para la zona. El tiempo entre sondeos es determinado por el valor de refresh en el registro SOA del dominio (típicamente, 12 horas o más).

El proceso DNS de sondeo se logra mediante el name server esclavo enviando una query al maestro de la zona pidiendo el registro SOA. Si el valor de serial de este registro es mayor que el valor actual mantenido por el name server esclavo, éste pide una full zone transfer (de aquí la importancia de mantener la disciplina y actualizar el número de serial cada vez que se cambia algo en alguno de los registros de la zona).

Las operaciones AXFR utilizan TCP en el puerto 53.

Incremental zone transfer

Transferir archivos de zona muy grandes puede llevar mucho tiempo y despedir cor ancho de banda y muchos recursos (especialmente si sólo cambió un registro). Por esto se introdujo la incremental zone transfer, que permite a los name servers esclavo y maestro transferir solo los registros que han cambiado.

El proceso es como en AXFR. El name server esclavo envía una query para el registro SOA del dominio al maestro de la zona cuando se alcanza el valor de refresh. Si el número de serial es mayor que el guardado actualmente por el esclavo, éste solicita una zone transfer e indica si es capaz o no de aceptar una IXFR. Si tanto el name server maestro como el esclavo soportan IXFR, esta transferencia se lleva a cabo, en caso contrario se realiza una AXFR. Las operaciones IXFR también utilizan TCP en el puerto 53.

IXFR sólo afecta el volumen de datos transferido; no tiene impacto en el tiempo que tardan en propagarse los cambios en archivos de zona.

NOTIFY

En el entorno altamente dinámico de Internet, un valor de refresh de 12 horas puede ser inaceptable. Se introduce un esquema en el cual un name server autoritativo (maestro o esclavo) enviará un mensaje NOTIFY a los name servers de la zona (definidos en los registros NS) cuando la zona es cargada o actualizada. Este mensaje indica que puede haber ocurrido un cambio en los registros del dominio. El name server que recibe el mensaje NOTIFY pedirá el registro SOA al maestro de la zona, y si el número de serial es mayor que el guardado actualmente, intentará una transferencia.

cia de zona usando AXFR o IXFR.

Actualización dinámica

El método clásico para actualizar los RR de una zona es editar manualmente el archivo de zona y luego reiniciar el name server para propagar los cambios. Cuando el volumen de cambios llega a cierto nivel, esto se vuelve operacionalmente inaceptable.

El santo gral de DNS es proveer un método de cambiar dinámicamente los registros de zona mientras el name server continúa respondiendo queries. Hay dos enfoques arquitecturales para resolver este problema: permitir actualización de los registros de zona en tiempo de ejecución desde una aplicación o fuente externa; o alimentar los registros de zona desde una base de datos que pueda ser actualizada dinámicamente.

El proceso DDNS (Dynamic DNS) forma el primer enfoque, donde los registros de zona se pueden actualizar desde una fuente externa. La limitación clave en esta especificación es que un nuevo dominio o zona no puede ser agregado dinámicamente. Todos los registros dentro de una zona existente pueden ser agregados, cambiados o eliminados, con la excepción del registro SOA que no puede ser agregado o borrado ya que esto esencialmente agrega o borraría la zona.

Se introdujo el término de maestro primario para describir el name server definido en el registro SOA para la zona. Cuando se actualizan dinámicamente los RR de una zona, es esencial actualizar solo un servidor, aún cuando pueda haber varios servidores maestros para la zona. Para resolver este problema, se debe elegir un servidor jefe. El servidor jefe, el maestro primario, no tiene características especiales fuera de ser el name server definido en el registro SOA y puede aparecer en la sentencia allow-update en el archivo de configuración named.conf.

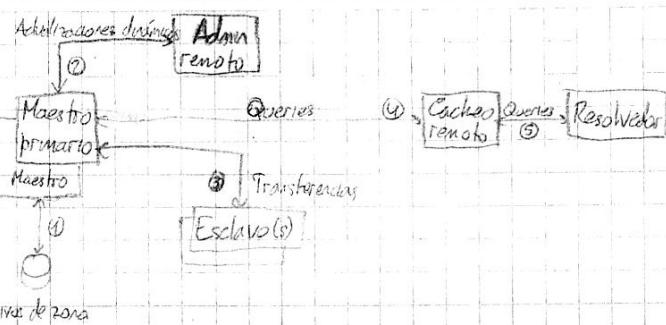
Seguridad en DNS

La operación DNS, el simple acto de correr un DNS, posibilita peligros potenciales de seguridad, ya que DNS es un servicio públicamente accesible.

El punto crítico al definir políticas y procedimientos de seguridad es entender qué necesita ser protegido, qué nivel de amenaza debe ser asegurado contra él y qué amenazas son aceptables. Esto depende de qué utilidad tiene el DNS que se quiere proteger.

Amenazas de seguridad

Para ser capaz de entender mejor las posibles amenazas y las posibles contramedidas, es primero necesario entender el flujo de datos normal en un sistema DNS. El siguiente diagrama ilustra el flujo, marcando las posibles fuentes de amenaza.



- ① Los amenazas locales son las más fáciles de prever; generalmente sólo se requiere una buena política de administración del sistema. Los archivos de zona y configuración deberían ser seguros, es decir, tener acceso limitado para leer y escribir y poseer copias de respaldo.
- ② Si se permiten las actualizaciones dinámicas (BIND las deniega por defecto), se corre el riesgo de actualizaciones no autorizadas, y de spoofing de direcciones IP (impersonar la fuente de la actualización). Un buen diseño de la arquitectura de red ayuda a disminuir el riesgo, así como utilizar TSIG y SIG(0). La amenaza se clasifica como servidor-servidor.
- ③ Si se mantienen name servers esclavos, se deben realizar zone transfers. Es posible utilizar varios name servers maestros en vez de servidores maestros-esclavos. Se corre el riesgo de spoofing de direcciones IP. Un buen diseño de la arquitectura de red ayuda a disminuir el riesgo, así como utilizar TSIG y TKEY. La amenaza es servidor-servidor.
- ④ Los cachés remotos pueden ser envenenados (contenidos corruptos que apuntan incorrectamente) por spoofing de IP, interceptación de datos y otros hacks. La solución es implementar DNSSEC completamente. Esta amenaza se clasifica como servidor cliente.
- ⑤ idén ④

Típos de DNS

Los servidores DNS asumen una amplia variedad de roles; un name server puede ser maestro para algunas zonas, esclavo para otras, y proveer caché o direccionamiento para otras. El rol de un name server es controlado por su archivo de configuración (named.conf). La combinación de parámetros globales y las zonas servidas determinan la funcionalidad completa de un name server.

Name servers: maestros

Una configuración DNS maestral de maestro de zona contiene uno o más archivos de zona para los cuales este DNS es autoritativo y que lee desde un sistema de archivos local. El término maestro se refiere a la ubicación del archivo de zona, en vez de alguna característica operacional.

El eslabón de maestro de zona se define con una sentencia en el archivo named.conf:

```
Zone "example.com" in {
    type master;
    file "/etc/named/example.com";
```

En este fragmento, zona "example.com" define la zona a la que se aplican las siguientes sentencias, tipo master define este DNS como maestro de zona para example.com, y file "master.com" define el nombre del archivo de zona conteniendo los RRs para example.com.

Es importante entender que un maestro de zona es un servidor que obtiene los datos de la zona de una fuente local, a diferencia de un esclavo de zona que los obtiene desde una fuente externa (típicamente el maestro o a través de una zone transfer). Esto significa que es posible tener cualquier número de maestros de zona para cualquier zona si tiene sentido operacional, pero hay que asegurarse de mantener sincronizados los archivos de zona en cada cambio.

Cuando un servidor DNS que es maestro para algunas zonas recibe una query para una zona para la que no es maestro ni esclavo, actuará como se lo configure. Si cachear está permitido y puede hacer queries recursivas, el servidor responderá completamente la solicitud o devolverá un error. Si cachear está permitido y sólo puede hacer queries iterativas, el servidor responderá completamente si tiene la respuesta en caché, devolverá una referencia, o devolverá un error. Si cachear no está permitido, el servidor devolverá una referencia o un error.

Un name server maestro puede indicar cambios en la zona a los esclavos mediante mensajes NOTIFY, lo que asegura que los cambios se propaguen rápidamente en vez de esperar el intervalo de refresh. En BIND se notifica por defecto a los name servers definidos en los registros NS de la zona.

Un maestro de zona puede estar escondido (sólo algunos de los esclavos conocen su existencia). No se requiere que un servidor maestro aparezca en los registros NS para el dominio, sólo que haya al menos 2 name servers que lo separen. Estos podrán ser cualquier combinación de maestro-esclavo, esclavo-esclavo o incluso maestro-maestro.

Name servers esclavos

Si bien es posible utilizar múltiples name servers maestros, pero cada archivo de zona con cambios debe ser copiado por todos los maestros. Además de este problema de sincronización, cada maestro debe ser recargado para utilizar los nuevos archivos de zona desactivando el name server por un tiempo. Para resolver este problema las especificaciones de DNS proveen el mecanismo de zone transfer, donde un name server, el esclavo, puede ser actualizado desde un maestro de zona mientras continúa respondiendo queries para la zona.

Un name server esclavo obtiene su información de un maestro de zona, pero responderá como autoritativo para aquellas zonas en las cuales se lo define como esclavo y tiene registros de zona válidos. Es imposible determinar

si una query fue respondida por el maestro de zona o por el esclavo. El acto de transferir la zona puede verse como delegar la autoridad para la zona al esclavo por el período de tiempo definido en el valor `expiry` del registro SOA, y por lo tanto permite al esclavo responder autoritativamente las queries.

El estado de esclavo se define con una sentencia en el archivo `named.conf`:

```
zone "example.com" {
    type slave;
    file "slave.example.com";
    masters { 192.168.23.17; }
```

Aquí, zone "example.com" indica la zona a la que se aplican las siguientes sentencias, type slave indica que este nombre server actuará como esclavo para example.com. La sentencia file "slave.example.com" es opcional y permite que los datos de la zona se guarden en un archivo específico. Si se recarga el name server, puede leer los datos de este archivo en vez de forzar una zone transfer, salvando tiempo y recursos. La sentencia masters { 192.168.23.17; } define la dirección IP de los name servers que tienen el archivo de zona maestro para la zona.

Un servidor esclavo intenta actualizar los registros de zona cuando se alcanza el valor de refresh del registro SOA. Si ocurre una falla, el esclavo intentará alcanzar al maestro periódicamente cada período de retry. Si un esclavo no pudo contactar al maestro cuando se llegó al tiempo de expiry, dejará de responder queries para la zona. El esclavo no usará datos que expiraron.

Comportamiento de un DNS esclavo

Los servidores esclavos responderán queries para el dominio como autoritativos mientras mantengan registros de zona válidos. Esta característica provee al usuario con una gran flexibilidad al registrar name servers para un dominio dado. Al registrar tales servidores, el único requisito es que los servidores listados respondan como autoritativos para queries para el dominio o zona. No es necesario definir al maestro de zona como uno de esos name servers; dos o más servidores esclavos cumplirán ese requerimiento. Esta flexibilidad permite al maestro de zona estar oculto del acceso público (maestro oculto). Esto permite que si en archivo de zona esclavo se vuelve corrupto a causa de un ataque malicioso, pueda ser rápidamente recuperado por el maestro mediante una zone transfer. Si el maestro no estuviera oculto y se volviera corrupto, los archivos de zona podrían tener que ser restaurados desde medios de respaldo, lo que podría tardar. Un maestro oculto es visible para los esclavos mediante el parámetro `master` en `named.conf`, pero no aparecerá en ningún registro NS para la zona. Todo name server (esclavo o maestro) que el usuario desee hacer visible debe estar definido usando

en registro NS en la zona.

Name servers de cacheo

Un name server de cacheo (cache DNS o resolvelor) obtiene información específica, en forma de un registro RR, sobre un dominio mediante queries a un name server autoritativo (maestro o esclavo) de una zona, para responder una query de un host/cliente, subsecuentemente guardando (cachendo) los datos localmente. En una solicitud posterior por los mismos datos, el servidor de cacheo responderá con los datos localmente almacenados del cache. Este proceso continuará hasta que el valor de TTL del registro expire, cuando el registro será descartado del cache. La próxima solicitud por este RR resultará en el resolvelor nuevamente mandando una query a un name server autoritativo para la zona. Los caches incrementan considerablemente el desempeño de DNS en PCs o hosts locales y también reducen la carga de la red al obtener una copia de datos frecuentemente necesarios y hacerla disponible muchas veces sin sobrecarga adicional.

Si el servidor de cacheo obtiene los datos directamente de un name server autoritativo de la zona, responderá una query como autoritativo, en caso contrario responderá como no autoritativo (cuando saca los datos de su cache).

Nota: la respuesta a una query es autoritativa bajo 3 condiciones: la respuesta proviene de un maestro de zona, la respuesta proviene de un esclavo de zona con datos de zona no expirados o la respuesta proviene de un servidor de cacheo que la recibió directamente de un maestro o esclavo de zona, no de su cache.

Si un name server de cacheo es recargado o reiniciado, los caches son usualmente borrados y el proceso comienza nuevamente. La única forma de que los datos de un RR sean removidos de una cache es o bien que su TTL expire, o el resolvelor se recargue.

Por defecto, en BIND los registros se cachean. Si un servidor provee cacheo entonces debe soportar queries recursivas, y éstas necesitan acceso a los root-servers. Esto se hace en el archivo named.conf:

options;

secure = yes; //pendiente. No sé si omifijo

allow recursion = 192.168.1.2/31; //Habilita la recursion (mejora la seguridad)

;

zone "." in { //toda zona no tiene en el cache

type hints; //elimina el dominio hints

; file "root.servers"; //archivo que contiene las A RR de los root-servers

Implicaciones del cacheo

Cachear o no cachear es una cuestión vital en el mundo DNS, ya que provoca sobrecarga de rendimiento. Además, como interactúa con redes externas, se corre el riesgo de envenenamiento de cache o corrupción a tra-

Véase de ataques maliciosos. Los usos más comunes de una configuración DNS de cacheo son:

- Un name server actuando como maestro o esclavo para algunas zonas y como un servidor de cacheo para todas las otras queries; un name server de propósito general.
- Un name server de sólo cacheo (resolvidor), típicamente utilizado para minimizar el acceso externo o compensar enlaces externos lentos.

Sin embargo, si un name server de propósito general es utilizado muchas veces por segundo en soporte de un sitio de alto volumen, el desempeño se vuelve un factor importante; en este caso, el cacheo será típicamente desactivado. Algunos administradores DNS, debido a los peligros relacionados con el cacheo, nunca permitirán cacheo en un name server que tenga zonas maestras o esclavas.

Forwarding (Proxy) name servers

Un forwarding (proxy, client o remote) DNS server es uno que direcciona todas las queries hacia otro DNS y cachea los resultados. Esto parece un ejercicio sin sentido. Sin embargo, un forwarding name server puede ser útil de las siguientes formas cuando el acceso a una red externa es lento o caro:

- El name server al que se direccionan las queries proveerá soporte para queries recursivas, resultando en una sola transacción query-respuesta. Si el name server local fuera sólo de cacheo y no direccionara queries, se producirían varias transacciones, incrementando la carga de la red.
- El forwarding DNS server local cacheará los resultados y, por lo tanto, proveerá respuestas rápidas para datos frecuentemente accedidos y eliminará tráfico externo innecesario.

Los forwarding servers pueden ser utilizados para facilitar la administración local: haciendo que cada host utilice un forwarding DNS, que a su vez pase todas las queries a un servidor externo. Si éste cambia, basta reconfigurar el servidor local en vez de todos los hosts. Además se puede utilizar como parte de una configuración stealth, para mejorar la seguridad.

La configuración de forwarding puede hacerse globalmente o por zona en el archivo named.conf:

options {

forwarders { 10.0.0.1, 10.0.0.2; };
 forward only;

zone "example.com" in {

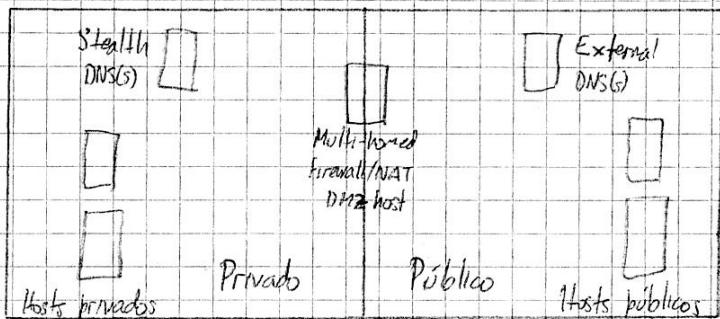
type forward;
 forwarders { 10.0.0.1, 10.0.0.2; };
 forward only;

La sentencia forwarders contiene las direcciones IP que serán utilizadas para direccionar (en roteación); la sentencia forward only fuerza todas las queries a ser direccionalas.

Stealth (Split) name servers

Un stealth server es definido como un name server que no aparece en ningún registro NS públicamente visible para el dominio. Los stealth servers son utilizados en configuraciones llamadas zonas demilitarizadas (DMZ) o split servers, y pueden ser definidos como tener las siguientes características:

- La organización necesita explorar servidores DNS para proveer acceso a sus servicios públicos como sitios web, e-mail, sitios FTP, etc.
- La organización no quiere que el mundo vea ninguno de sus hosts internos ya sea por intercepción (query o zone transfer) o en el caso que el servicio DNS se vea comprometido.



Los servidores externos o públicos están configurados para proveer respuestas sólo autoritativas y no cachean (no quieren recursivas). En este caso, cada vez desperdicia recursos y es una posible fuente de corrupción. El archivo de zona utilizado por estos servidores contendrá sólo los sistemas o servicios que el usuario necesita hacer visible externamente, como el registro SOA, los registros NS de los name servers públicos, los registros MX para servicio de mail y registros A para sitios web públicos, servicios FTP, etc.

Los zone transfers pueden permitirse entre los name servers públicos cuando se necesite, pero no deberían ocurrir transferencias desde el stealth server. Esta clara separación entre los hosts público x privado de la red es requerida ya que si el name server público es comprometido, la inspección de los archivos de configuración o de zona no debería proveer información que describa alguna parte de la red oculta.

El stealth server puede ser configurado para hacer visible los servicios internos y externos, proveer queries recursivas y todo tipo de otros servicios. Un archivo de zona para un stealth server contendrá información sobre los detalles públicos, y también sobre la parte privada de la red.

Claramente, los usuarios internos deben cruzar el perímetro en algún momento para acceder servicios externos, incluyendo servicios DNS. Hay 2 posibles soluciones a este problema:

- La solución de Firewall en la red los sistemas internos, incluido el servidor DNS, tienen permitido enviar y recibir los externamente en una base transacción por transacción.

- Utilizar la clásica view de BIND para proveer soporte de cacheo y queries recursivas para la red interna en el servidor DNS público, mientras se continúa desregalar estos servicios a los usuarios externos y sin exponer la estructura de la red interna. Esta solución no elimina la necesidad de un firewall para el tráfico no perteneciente a DNS.

Name servers sólo autoritativos

El término de name server sólo autoritativo es normalmente utilizado para describir dos conceptos:

- El name server entregará respuestas autoritativas; es un maestro o esclavo de zona para uno o más dominios.
- El name server no cachea.

Hay 2 configuraciones en las que se utilizan típicamente servidores sólo autoritativos:

- Como servidores públicos en una configuración DNS stealth usada para proveer seguridad en el perímetro.
- Como name servers de alto desempeño, por ejemplo, root-servers, TLD servers o servidores para sitios de alto volumen.

No es posible desactivar completamente el cacheo en BIND, pero la sentencia de recursión inhibe efectivamente el cacheo en el archivo named.conf:

```
options {
    recursion no;
};
```

Firewall/Iptables

Una persona maliciosa que obtiene acceso a una computadora podría activar contraseñas del sistema, o explotar los bugs o el comportamiento de ciertos programas para obtener una cuenta válida en ese host. Una vez que son capaces de logearse en el host, podrían acceder a información importante. En un marco comercial, robar, borrar o modificar información como planes de marketing, detalles de nuevos proyectos o bases de datos de información de clientes puede causar daños importantes a la compañía.

La forma más segura de evitar tales daños es prevenir personas no autorizadas de ganar acceso al host desde la red. Aquí es donde entran los firewalls.

Métodos de ataque

Es importante entender la naturaleza de los ataques potenciales en seguridad computacional.

Acceso no autorizado

Esto significa simplemente que personas que no deberían tener permitido el uso de servicios de una computadora son capaces de conectarse a ellos y usarlos. Hay varias formas de evitar este ataque especificando cuidadosamente quién puede tener acceso a estos servicios. Se puede prevenir el acceso desde la red a todos, salvo los

Usuarios es perfeccionados.

Explotación de debilidades conocidas en programas

Algunos programas y servicios de red no fueron originalmente diseñados con fuerte seguridad como objetivo y son inherentemente vulnerables a ataques. La mejor manera de protegerte contra este tipo de ataques es desactivar cualquier servicio vulnerable o encontrar alternativas. Una buena manera de empezar es sólo instalar, ejecutar y explotar servicios absolutamente necesarios. También es útil tener bases de datos de bugs y listas de parches, y mantener los sistemas actualizados.

Negación de servicio

Los ataques de negación de servicio causan que el servicio o programa deje de funcionar o previene que otros lo usen. Esto puede ser hecho en la capa de red enviando paquetes cuidadosamente diseñados y maliciosos que causan fallos en las conexiones de red; o en la capa de aplicación, donde comandos de aplicación cuidadosamente diseñados son entregados a un programa causando que se vuelva muy ocupado o deje de funcionar.

Prevenir tráfico de red sospechoso de llegar a los hosts y prevenir comandos y solicitudes de programas sospechosos son las mejores formas de minimizar el riesgo de un ataque de negación de servicio.

Spoofing

Este tipo de ataque involucra un host o aplicación haciéndose pasar por otro. Tipicamente el host atacante finge ser un host inocente forjando direcciones IP en tráfico de red. Para protegerte, se debe verificar la autenticidad de los paquetes y comandos (una combinación de filtrados y servicios proxy funciona bien), prevenir el ruteo de paquetes con direcciones de fuente inválidas, y usar sistemas operativos con mecanismos de control de conexión impredecibles, como números de secuencia TCP y asignación de puertos dinámica.

Eavesdropping

Este es el tipo de ataque más simple. Un host se configura para "escuchar" y capturar datos que no le pertenezcan, poniendo su interfaz de red en modo "promiscuous" y monitoreando todos los paquetes que atraviesan el segmento de red. Programas de eavesdropping cuidadosamente escritos pueden obtener nombres de usuario y contraseñas desde conexiones de logeo de usuario. Las redes inalámbricas son especialmente vulnerables ya que no requieren acceso físico, solo proximidad.

Para protegerte contra este tipo de amenaza, se debe evitar el uso de tecnologías de broadcast de red y enfocar el uso de encriptación de datos.

El firewall de IP es muy útil previniendo o restringiendo accesos no autorizados, negación de servicio en capa

de red y ataques de IP spoofing. No es muy útil para evitar explotación de debilidades en servicios de red o programas maliciosos y eavesdropping.

Firewall

Un firewall es un host confiable y robusto que actúa como escudo de batalla entre un grupo de redes (usualmente una red pública y una red privada). Todo el tráfico de red entre las redes afectadas se rutea por el firewall. El host de firewall se configura con un conjunto de reglas que determina qué tráfico de red se dejará pasar y cuál se bloqueará (sin respuesta) o rechazará (con respuesta).

Los firewalls pueden ser construidos de varias maneras. La más sofisticada involucra varios hosts separados y se conoce como red de perímetro o zona desilitarizada (DMZ). Los hosts actúan como "filtros" para permitir que solo cierto tipo de tráfico de red pase, y entre esos filtros residen servicios de red como correo o un servidor proxy HTTP. Esta configuración puede ser muy segura y permite un gran control sobre quién se puede conectar desde adentro hacia afuera y viceversa.

Sin embargo, en muchos casos se construyen firewalls que además proveen otros servicios. Estos son menos seguros ya que si alguien explota una debilidad en uno de los servicios externos que corre el firewall, toda la seguridad de la red ha sido penetrada. El atacante podría modificar las reglas del firewall para permitir más acceso y deshabilitar avisos que podrían notificar al administrador de la red sobre actividad sospechosa. La ventaja de estos firewalls es que son más baratos y fáciles de manejar que una DMZ.

El núcleo Linux provee un rango de características incorporadas que le permiten funcionar como un firewall IP. La implementación de red incluye para procesar paquetes IP de varias maneras (el subsistema netfilter) y provee un mecanismo de espacio de usuario para configurar qué clase de reglas se desea poner (el comando iptables).

Filtrado IP

El filtrado IP es un mecanismo que decide qué tipo de paquetes IP serán procesados normalmente y cuáles serán descartados o rechazados. Por descartado queremos decir que el paquete será borrado y completamente ignorado, como si no se hubiera recibido. Por rechazado queremos decir que el firewall envía una respuesta ICMP al emisor indicando la razón de que el paquete se ha rechazado. Se pueden aplicar varios criterios para determinar los paquetes que se deben filtrar, como el tipo de protocolo, el número de puerto, el tipo de paquete, la dirección de origen del paquete, la dirección de destino del paquete, etc.

Es importante entender que el filtrado IP es un mecanismo de control de red. Esto significa que no sabe nada sobre la

aplicación que usa las conexiones de red, sólo sobre las conexiones en sí mismas. Para complementar esto se deberían utilizar servidores proxy que entiendan la aplicación.

iptables

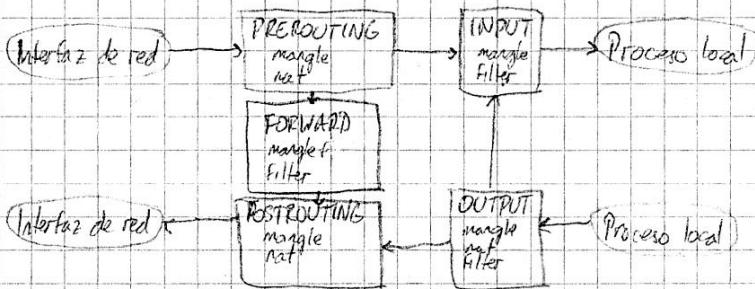
La arquitectura iptables agrupa reglas de procesamiento de paquetes de red en tablas por función (filtrado de paquetes, NAT, ...), cada una de las cuales tiene cadenas (secuencias) de reglas de procesamiento. Las reglas consisten en matchers (usados para determinar a qué paquetes se aplica la regla) y objetivos (que determinan qué se hace con los paquetes que matchean).

iptables define 5 "puntos de enganche" en los canales de procesamiento de paquetes del núcleo: PREROUTING, INPUT, FORWARD, POSTROUTING y OUTPUT. Las cadenas mencionadas se enganchan a estos puntos de enganche; se puede agregar una secuencia de reglas para cada una de ellas.

Cada regla es una línea que el núcleo lee para saber qué hacer con un paquete. Si todos los matchers se cumplen, entonces se ejecuta la instrucción objetivo.

Una regla tiene una estructura como esta: #iptables [! tabla] comando [match] [objetivo/salto]

Flujo de paquetes



Las cajas representan las cadenas de iptables, y dentro de cada una hay una lista de las tablas que tiene.

- FORWARD: permite procesar paquetes que fluyen en una computadora entrando por una interfaz y saliendo por otra (la primera y última decisión de routing).

- INPUT: permite procesar paquetes justo antes de que se entreguen a un proceso local.

- OUTPUT: permite procesar paquetes justo después de que los genere un proceso local.

- POSTROUTING: permite procesar paquetes justo antes de que salgan de una interfaz de red.

- PREROUTING: permite procesar paquetes justo después de que llegan desde una interfaz de red.

Tablas

La opción -t especifica que tabla utilizar. iptables viene con 3 tablas incorporadas: filter, mangle y nat. Por defecto se emplea la tabla filter.

- nat: se emplea principalmente para el protocolo NAT. Los paquetes que son filtrados por esta tabla acaban con sus IPs modificadas, de acuerdo con las reglas de la tabla. Sus cadenas incorporadas son PREROUTING, OUTPUT, y POSTROUTING.
- mangle: se emplea principalmente para retoque paquetes. Entre otras cosas, se puede cambiar el contenido de diferentes paquetes y el de sus cabeceras (como cambiar los campos TTL o ToS). Tiene las 5 cadenas incorporadas.
- filter: se debería emplear exclusivamente para filtrar paquetes. Por ejemplo, se pueden desear bimbo a un registro de sucesos, aceptar o rechazar paquetes sin problemas. Sus cadenas incorporadas son FORWARD, INPUT y OUTPUT.

Comandos

Un comando le indica a iptables que hacer con el resto de la regla que enviamos al analizador. Normalmente se debe indicar añadir o borrar algo en una tabla determinada.

- A(-add) chain rule: agrega rule a chain, al final.
- D(-delete) chain [index rule]: borra la regla en posición index o el match rule de chain.
- R(-replace) chain index rule: reemplaza la regla en la posición index de chain por la regla rule.
- I(-insert) chain [index]: inserta rule en chain, en la posición index o al principio. -T chain [trueum] rule -specification
- L(-list) [chain]: lista las reglas de chain, o de todas las cadenas.
- F(-flush) [chain]: borra todas las reglas de chain, o de todas las cadenas.
- Z(-zero) [chain]: pone en 0 los contadores de chain, o de todas las cadenas.
- N(-newchain) chain: crea la cadena chain.
- X(-delete-chain) [chain]: borra la cadena chain, o todas las cadenas (creada por usario).
- P(-policy) chain target: define la política por defecto de chain como target.
- E(-renumber-chain) chain newchain: renombra chain como newchain.

También hay opciones que pueden usarse con ciertos comandos.

- v(-verbose): ofrece una salida detallada. Se usa con -L, -A, -I, -D y -R.
- x(-exact): muestra los números exactos para contadores, en vez de abreviaciones por defecto. Se usa con -L.
- n(-numeric): muestra valores numéricos en direcciones IP y puertos, en vez de nombres de dominio, red o aplicación. Se usa con -L.
- l(-line-numbers): empleado con -L, cada regla se listará con su número de línea.
- c(-set-counters) packet bytes; combinado con -A, -I o -R, pone el contador de paquetes en bytes y el contador de bytes.

tes en bytes.

--M(--match): se emplea para cargar un nuevo iptables. Se usa con todos los comandos.

Matches

Podemos clasificar los matches en 5 categorías: genéricos, TCP, UDP, ICMP y especiales.

Matches genéricos

Un match genérico es un tipo de condición que está siempre disponible, sea cual sea el protocolo con el que se trabaje y no se necesitan otros específicos para emplearlos. El símbolo ! indica una negación de los argumentos (todos menos los indicados).

- ![-proto] [!] proto: se emplea para comprobar la presencia de los protocolos TCP(6), UDP(17) o ICMP(1). También se puede utilizar el comando ALL(0).

- ![-src, --source] [!] addr [!mask]: comparación del origen, basada en la dirección IP de origen de los paquetes. Se pueden especificar máscaras para trabajar con rangos de direcciones.

- ![-dst, --destination] [!] addr [!mask]: comparación del destino. Mismas formas que src.

- ![-in-interface] [!] in: empleada para reconocer a través de que interfaz proviene un paquete entrante. Se puede agregar + al final de in para reconocer interfaces que empiecen con in.

- ![-out-interface] [!] out: empleada para paquetes que están a punto de abandonar la interfaz de salida. Mismas formas que in.

- [!] -f(-fragment): empleada para checar la segunda(y posterior) parte de un paquete fragmentado.

Matches TCP

Estos matches son específicos del protocolo y sólo están disponibles al trabajar con paquetes y flujo TCP. Para emplearlos se debe indicar -p tcp en la línea de comandos a la izquierda de ellos.

--sport(--source-port) [!] port[:port]: matchea cuando el puerto TCP de origen es igual a port o está en el rango especificado. El puerto se puede nombrar por servicio o número. Si se omite el primer puerto del rango, se asume 0.

--dport(--destination-port) [!] port[:port]: similar a sport pero comparando el puerto de destino.

--tcp-flags [!] mask comp: checar las banderas mask, y matchea sólo si las banderas comp son activadas. Se reconocen las banderas SYN, ACK, FIN, RST, URG, PSH, ALL, NONE. Los argumentos se listan separados con comas.

- [!] !syn: equivalente a --tcp-flags SYN,RST,ACK syn. Con esto se consigue bloquear cualquier intento de conexión desde el exterior.

--tcp-option [!] num: matchea si la opción TCP num está seteada.

--mss value[:value]: matchea paquetes SYN y ACK cuando el valor del campo MSS es igual a value o está dentro del rango especificado.

Matcheos UDP

Estos matcheos sólo trabajan con paquetes UDP y se debe indicar -p UDP para que estén disponibles:

--sport[!-source-port] [!] port[:port]: igual que en TCP.

--dport[!-destination-port] [!] port[:port]: igual que en TCP.

Matcheos ICMP

Estos matcheos sólo se utilizan con paquetes ICMP y se debe indicar -p ICMP para habilitarlos.

--icmp-type[!] type[:code]: matchea el tipo ICMP con el nombre del tipo o su valor numérico.

Matcheos especiales

Estos matcheos se deben cargar específicamente con la opción -m o --match.

-limit: este match se utiliza para limitar las veces que una regla puede ser alcanzada en un período de tiempo dado útil para minimizar DDoS). La opción [k] --limit val/interval establece cuantas veces se le permite actuar al match por unidad de tiempo. La opción --limit-burst num: especifica el número máximo de paquetes que coinciden con el match en un tiempo determinado.

-mac: se usa para comparar paquetes en función de su dirección MAC de origen. La opción --mac-source/mac permite especificar la dirección, con el formato xx:xx:xx:xx:xx:xx.

-mport: se emplea para especificar puertos múltiples y rangos de puertos múltiples. Si este match no existiera se tendría que escribir varias reglas del mismo tipo simplemente para matchear puertos que no se pueden especificar mediante un rango. No se pueden combinar la configuración anterior con la múltiple. La opción --source-port list matchea con los puertos de origen indicados (máximo 15). La opción --destination-port list matchea con los puertos de destino indicados.

La opción --port matchea con los puertos de origen/destino indicados, solo se aceptan paquetes que lo atraviesan y se destinan al mismo puerto.

-state: se emplea conjuntamente con el código de seguimiento de conexiones del núcleo. El match de estado accede a la máquina de seguimiento de conexiones y averigua en qué estado se encuentra el paquete. La opción --state list le indica al match state que estados deben ser combinados. Hay 4 estados que pueden ser utilizados: INVALID, ESTABLISHED, NEW, RELATED. INVALID implica que el paquete no está asociado a ningún flujo o conexión conocida y que

puede contener datos o cabeceras erróneas. ESTABLISHED indica que el paquete pertenece a una conexión ya establecida, totalmente válida y que "ha visto" un flujo de paquetes en ambas direcciones. NEW significa que el paquete ha creado o está creando una nueva conexión, o que está asociado a una conexión que todavía no "ha visto" paquetes en ambas direcciones. RELATED es para paquetes que empiezan una nueva conexión pero están asociados a otra conexión ya establecida.

Targets/Saltos

Cuando el match de una regla encuentra un paquete que coincide con las condiciones que impone, se recorre al target/salto donde se le indica a la regla qué debe hacer con ese paquete.

Saltos

La orden de salto se ejecuta de la misma manera que la orden de target, excepto en que el salto necesita que exista una cadena dentro de la misma tabla a la que pertenece. Los paquetes saltarán a la otra cadena y serán filtrados por ella. Si algún paquete llega al final de la cadena y no ha hecho una concordancia, volverá a la cadena original justo después de la regla que originó el salto y seguirá pasando por el resto de las reglas de esta cadena. Por el contrario, si alguna de las reglas de la nueva cadena acepta los paquetes, estos efectuarán el target/salto de la regla y seguirán su curso por el resto de las cadenas, sin pasar por el resto de las reglas de la cadena original.

Targets

Los targets especifican la acción a ejecutar con el paquete en cuestión. Así, se podrá aceptar o desechar el paquete, u otras acciones. Algunos targets harán que el paquete deje de atravesar una cadena específica y las superiores a ella, otros podrán efectuar alguna acción sobre el paquete, después de la cual el paquete continuará pasando por el resto de reglas.

Target ACCEPT

Este objetivo no necesita ninguna opción adicional. En cuanto la especificación del match es satisfecha por el paquete y se indica ACCEPT como target, la regla se acepta y el paquete no continuará atravesando la cadena actual, ni cualquier otra de la misma tabla. Sin embargo, un paquete aceptado por una cadena podría todavía atravesar las cadenas de otras tablas y ser desechar en ellas. Se indica -j ACCEPT.

Target DNAT

DNAT se emplea para efectuar traducciones de direcciones de red de destino, es decir, reescribir la dirección IP destino de un paquete. Con este objetivo en la regla, en cuanto un paquete coincide con el match, él y todos los paquetes posteriores a ese mismo flujo de datos verán modificada su dirección de destino y serán redirigidos a la

red/bastidor dispositivo adecuado. Esto puede ser útil cuando se tiene un host ejecutando el servidor dentro de una red local, pero sin una IP real que ofrecerle y que sea válida en Internet; se puede indicar al firewall que cuando llegan paquetes dirigidos a su puerto puerto (HTTP) los reenvíe hacia el servidor web real dentro de la red local. El target DNAT solo está disponible en las cadenas PREROUTING y OUTPUT, así como cadenas-salto a partir de ellas.

La opción `--to-destination ip-range [port-range]` le indica al mecanismo DNAT que IP de destino establecer en la encuesta IP y dónde enviar los paquetes que concuerden con el filtro. También se puede especificar un puerto al que será redirigido el tráfico (solo cuando -p es tipo UDP).

Target DROP

El objetivo DROP desechará paquetes y no se gasta más trabajo de procesador en ellos. Un paquete que llegue a una regla, coincida con el patrón de búsqueda del match y sea desechar, será inmediatamente bloqueado. Si se efectúa la acción DROP a un paquete dentro de una subcadena, este paquete ya no será procesado en ninguna de las cadenas principales de ninguna tabla; el paquete estará "muerto" de inmediato, y el objetivo no enviará información alguna en ninguna dirección ni responderá a intermediarios como los routers.

Target LOG

El objetivo LOG se ha diseñado especialmente para registrar información detallada sobre los paquetes, como la mayoría de las cabeceras IP y otros datos interesantes. Este target es excelente para depurar y afinar los conjuntos de reglas, datos que se pueden ver a dónde van y qué reglas se aplican a los paquetes. Las opciones del target LOG son `--log-level level`, `--log-prefix prefix`, `--log-tcp-sequence`, `--log-tcp-options` y `--log-ip-options`.

Target MASQUERADE

El objetivo MASQUERADE se usa básicamente de la misma forma que el objetivo SNAT, pero sin requerir ninguna opción `--to-source`. La razón es que el target MASQUERADE se creó para trabajar con conexiones de tipo de conexión DHTCP, que obtienen direcciones IP dinámicas al conectar a la red.

Enmascarar una conexión significa que se establece la dirección IP utilizada por un puerto dentro de una interfaz específica. Además, en el target MASQUERADE las conexiones se pierden en la interfaz se acelera, evitando tener datos viejos de conexión que ocupen la memoria de seguimiento de conexiones. La opción `--to-ports port` se utiliza para establecer el puerto de origen de los paquetes salientes. Se puede especificar un rango de puertos (los extremos se separan con un -).

Target QUEUE

El objetivo QUEUE se emplea para poner en cola los paquetes dirigidos a programas y aplicaciones del espacio de usuario.

Target REDIRECT

El objetivo REDIRECT sirve para redirigir paquetes y flujo hacia la máquina. Los paquetes generados localmente son enviados a la dirección 127.0.0.1. La opción --to-ports port especifica el puerto o rango de puerto de destino que se debe usar.

Target REJECT

El objetivo REJECT funciona básicamente igual que el objetivo DROP, aunque en este caso si se devuelve un mensaje de error al host que envió el paquete bloqueado. La opción --reject-with answer le indica a REJECT qué respuesta devolver al host que envía el paquete que se rechaza. answer puede ser icmp-net/host/port/proto-unreachable, icmp-net/host/prohibited, echo-reply y tcp-reset.

Target RETURN

El objetivo RETURN hará que el paquete que está atravesando una cadena pare allí donde se encuentre con el target; si está en una sub-cadena, el paquete continuará atravesando la cadena superior donde la dejó; si es una cadena no se pondrá, se ejecutará la política por defecto sobre el paquete.

Target SNAT

El objetivo SNAT se emplea para efectuar las traducciones de dirección de red de origen, lo cual implica que este objetivo reescribirá la dirección IP de origen en la cabecera IP del paquete. Así, podemos escribir una regla SNAT que cambiará la dirección IP de origen de todos los paquetes que salgan de la red local por la dirección IP de origen de nuestra conexión a Internet.

El objetivo SNAT sólo es válido en la tabla nat y dentro de la cadena POSTROUTING. Sólo el primer paquete de una conexión es modificado por SNAT, después del cual todos los paquetes pertenecientes a la misma conexión seguirán modificados de la misma manera que el primero. La opción --to-source ip[:port] se emplea para especificar qué dirección de origen debe utilizar el paquete. Se puede indicar un rango de direcciones, así como un puerto o rango de puertos para el puerto de origen.

Target TOS

El objetivo TOS se emplea para establecer el campo Type of Service de la cabecera IP. La opción --set-bit-index le dice al programador de modificaciones TOS que valor (hexadecimal) establecer en los paquetes que