Resumen 1er parcial de Redes y Comunicación de Datos II

5.1 ASPECTOS DE DISEÑO DE LA CAPA DE RED

Objetivos

La capa de red se encarga de llevar los paquetes desde el origen hasta el destino. Para lograr su cometido, la capa de red debe conocer la topología de la subred de comunicación y elegir las rutas adecuadas a través de ella. Por último, cuando el origen y el destino están en redes diferentes, ocurren nuevos problemas. La capa de red es la encargada de solucionarlos.

Servicios proporcionados a la capa de transporte

Dados los objetivos de la capa de red, existe una discusión entre el tipo de conexión que debe brindar. Por una lado esta la comunidad de internet que apoya un servicio no orientado a la conexión, donde los host realicen el ordenamiento de paquetes y el control de flujo. El otro bando, representado por las compañias telefonicas dice que la capa de red debe proporcionar un servicio orientado a la conexión y confiable. Ejemplos de estas dos posturas son Internet y ATM.

Comparación entre servicios OC y no OC

Por un lado, el servicio no orientado a la conexión funciona enviando **datagramas** a travez de una subred que se conoce como **subred de datagramas**. Estos se envian y enrutan de forma independiente entre si. Los routers deciden la ruta que debe tomar cada datagrama por medio de un **algoritmo de enrutamiento**. Por otro lado el servicio orientado la conexión funciona enviando los paquetes a travez de **circuitos virtuales** establecidos al inicio de la conexión.

Asunto	**Subred de datagramas**	**Subred de circuitos virtuales**
Configuración del circuito	No	Requerida
Direccionamiento	Cada paquete contiene la dirección de origen y destino	Cada paquete contiene un numero de CV(corto)
Información de estado	Los enrutadores no contienen información de estado de las conexiones	Cada CV requiere espacio de tabla del enrutador por conexión
Enrutamiento	Cada paquete se enruta de forma independiente	Se escoge al establecer el CV.
Efecto de fallas del enrutador	Ninguno, salvo paquetes perdidos en la caida	Terminan todos los Cvs que pasan por el enrutador
Calidad del servicio	Difícil	Fácil. Se asignan recursos por adelantado.
Control de congestión	Difícil	Fácil. Se asignan recursos por adelantado.

5.2 ALGORITMOS DE ENRUTAMIENTO

El algoritmo de enrutamiento es aquella parte del software de la capa de red encargada de decidir la línea de salida por la que se transmitirá un paquete de entrada. Si la subred usa datagramas de manera interna, esta decisión debe tomarse cada vez que llega un paquete de datos, si la subred usa circuitos virtuales internamente, las decisiones de enrutamiento se toman sólo al establecerse un circuito virtual nuevo.

Es importante hacer una distinción entre el enrutamiento, que es el proceso consistente en decidir cuales rutas utilizar, y el reenvío, que consiste en la acción que se toma cuando llega un paquete(buscar en las tablas de enrutamiento la línea de salida por la cual se enviara el paquete).

Hay ciertas propiedades que todo algoritmo de enrutamiento debe poseer: exactitud, sencillez, robustez, estabilidad, equidad y optimización. La robustez indica que una red pueda permanecer en funcionamiento sin fallas a nivel de sistema durante años, pese a que halla fallas en los host, se caigan routers, etc. Las propiedades de equidad y optimización a veces son contradictorios, se debe encontrar un punto medio entre los dos.

Los algoritmo de enrutamiento se dividen en dos clases, algoritmos adaptativos y no adaptativos. Los primeros cambian sus decisiones de enrutamiento para reflejar los cambios de topología y, por lo general, también de tráfico. Los segundos definen las rutas que se usaran por adelantado, fuera de línea (enrutamiento estático).

Principio de optimización

Este postulado establece que si el enrutador B esta en la ruta óptima entre A y C, entonces la ruta óptima entre B y C también está en la misma ruta.

Como consecuencia de este postulado, podemos ver que el grupo de rutas óptimas de todos los orígenes a un destino dado forman un árbol con raíz en el destino. Tal árbol se conoce como **árbol sumidero** o **árbol divergente**. Este no es único, ya que pueden existir otros árboles con als mismas longitudes de rutas.

Es importante aclarar que cuando se habla de la ruta óptima es con respecto a una métrica dada. Esta puede ser: menor cantidad de saltos, menor distancia física, retardo medio de encolamiento, ancho de banda, combinaciones de varias métricas, etc.

Enrutamiento por la ruta más corta

Este método utiliza el algoritmo de la ruta mas corta en un grafo (Dijkstra por ej.) Donde los nodos son routers y las aristas son enlaces dirigidos, que llevan como peso la métrica elegida.

Inundación

Este método consiste en enviar cada paquete que llega por todas las lineas de salida excepto por la que llego. La inundación genera infinita cantidad de paquetes si no se toman medidas para evitarlo.

Una posible estrategia es colocar en el encabezado de cada paquete un contador que se disminuya con cada router que atraviesa. Al quedar en 0 el paquete no se retransmite. Este contador se debe iniciar con la distancia del origen al destino, o si se desconoce, la distancia del peor caso.

Otra técnica alternativa es llevar un registro de los paquetes difundidos para evitar enviarlos una segunda vez. Para ello cada router de origen pone un numero de secuencia en los paquetes que envía. Además, cada router lleva una lista donde el i-esimo numero es el ultimo numero de secuencia de paquete retransmitido desde el router i.

Una variación de la inundación es la **inundación selectiva.** En este algoritmo, los enrutadores solo envían cada paquete solo por las lineas que van aproximadamente en la dirección correcta.

Enrutamiento por vector de distancia (Bellman-Ford)

En este método, en cada nodo(router) de la red se almacena una tabla de 3 columnas. En la primera se coloca el destino, en la segunda el retardo medido para alcanzar ese destino y en la tercera la linea de salida por la que se debe enviar el paquete. Entonces cuando llega un paquete con destino al router i-esimo, se busca en la tercera columna de fila i porque linea se lo debe enviar.

Lo importante de este método es la forma en que se arman estas tablas. Cada cierto tiempo, los routers miden la distancia a cada uno de sus vecinos utilizando alguna métrica (si son saltos por ej es igual a 1, si es retardo se hace un echo y se mide el tiempo, etc). Con estas mediciones reemplazan los valores de la tabla correspondientes a las filas de sus vecinos. Luego intercambia su tabla con la de todos sus vecinos. Entonces para cada fila correspondiente a un router que no es vecino, revisa las mediciones de las tablas de sus vecinos y elige la menor. En esa fila entonces coloca como retardo la suma entre el tiempo hasta ese vecino y el retardo que informo el vecino. Y como linea de salida coloca el vecino.

Este algoritmo sin embargo presenta una falla llamada el problema de las cuentas hasta infinito. Se puede resumir diciendo que las buenas noticias viajan rápido por la red, mientras que las malas viajan lento. (ver del libro.. es largo de explicar..)

Enrutamiento por estado del enlace

El algoritmo consiste en los siguientes pasos:

1. Descubrir a sus vecinos y conocer sus direcciones Ips:

   Para alcanzar el objetivo, cada router envía un paquete HELLO por cada linea de salida punto a punto. Se espera que del otro lado contesten identificandose. Un problema surge cuando hay varios routers conectados a una misma LAN. Para solucionarlo se modela la LAN como si fuera un nodo mas al que se conectan todos los routers.

2. Medir el retardo o costo para cada uno de sus vecinos:

   Para realizar esto se envía un paquete ECHO por cada linea y se mide el tiempo que tarda en ir y ser contestado, luego se divide el tiempo por dos. Se pueden realizar varias mediciones de estas y sacar un promedio para mejorar el calculo. Una cuestión importante es si considerar o no el tiempo de retraso. Considerarlo a veces da mejores resultados, pero puede ocasionar fluctuaciones.

3. Construir un paquete que indique todo lo aprendido:

   El paquete consiste en un encabezado formado por el nombre del router de origen, un numero de secuencia de 32 bits, un numero que representa la edad, y la lista de vecinos del router origen con el retardo que hay del origen a cada vecino.

4. Enviar este paquete a todos los enrutadores:

   Se utiliza inundación. Para controlarla se mira el numero de secuencia de los paquetes. Si a un router llega un paquete con numero de secuencia menor al ultimo que llego del mismo destino, simplemente se descarta por ser mas viejo. En otro caso se almacena y difunde por el resto de las lineas de salida. Además se utiliza el campo de edad para hacer que la información de los routers se descarte pasado cierto tiempo. Como se envían paquetes de estado cada intervalos pequeños de tiempo, esto solo sucede si el router se cae, o si se

pierden muchos paquetes consecutivos del mismo router. Dentro del router se almacenan estos datos en tablas que tienen los campos: origen, secuencia, edad, banderas de ACK, banderas de reenvío y datos. Las banderas son para controlar a que vecinos se les confirma la recepción y a quienes se les reenvía.

5. Calcular la ruta mas corta a todos los demás enrutadores.

   Se calcula Dijkstra localmente y se almacenan los caminos mas cortos a cada destino en tablas dentro del router.

Enrutamiento jerarquico

Cuando el numero de nodos crece, el almacenamiento de las tablas de dirección se puede volver un problema. Una forma de mitigar este problema es dividir la subred en varias regiones donde cada router perteneciente a una región solo conoce el camino hacia los nodos de su misma subred, salvo algunos que funcionan de puerta de salida hacia otra región. Cuando un router de la región A quiere enviarle a otro de la región B, el primero envía el paquete al router de salida de su región, este se lo envía al de salida de la región B y este termina enviándoselo al router destino.

Enrutamiento por difusión.

A veces un router quiere enviar un paquete a todos los nodos de su subred. Para lograr esto existen varios algoritmos:

- Enviar el paquete a todos los nodos de la subred: El mas simple pero desperdicia ancho de banda, además cada nodo debe conocer la dirección del resto de routers.
- Inundación: También es simple, pero desperdicia ancho de banda.
- Enrutamiento multidestino: en este método cada paquete lleva una lista de destinos. Cada vez que un paquete llega a un router se reenvía solo por las lineas necesarias(las que estén dentro del mejor camino a algún destino) una copia donde la lista de destino solo contiene los destinos que se alcanzan por esa linea. Llega un punto en que la lista solo contiene un destino, y el paquete se maneja como un paquete común.
- Enrutamiento por árbol de expansión: Cada router almacena un árbol de expansión del grafo de la subred. De esta forma envía paquetes solo por los enlaces que pertenecen al árbol.
- Reenvío por ruta invertida: Cuando llega un paquete a un router, este revisa si la linea por la que llego es la mejor la ruta hacia el origen. Si es así se reenvía por todas las lineas menos por la que llego. En otro caso se elimina.

Enrutamiento por multidifusión

Para realizar multidifusión de manera eficiente y segura, cada router almacena un árbol de expansión de toda la subred. Cuando se quiere enviar un paquete a cierto grupo, primero se poda el árbol eliminando todos los enlaces que no conduzcan a un router del grupo, luego el paquete se envía por el árbol resultante.

Cuando los router no conocen toda la topología de la red (por ej cuando se usa vector distancia) lo que se hace es utilizar un algoritmo que consiste se basa en el algoritmo de ruta invertida. Cuando a un router llega un mensaje para el grupo A y no tiene ningún router de ese grupo y ningún router de salida, este contesta con un paquete PRUNE. Cuando un router recibe PRUNE de todos sus vecinos, le envía PRUNE al nodo desde el cual recibió el mensaje. De esta forma se van podando los arboles de expansión de forma recursiva. El problema con este algoritmo es que cada router debe almacenar

varios arboles de expansión, uno por cada grupo. Una posible solución a esto es utilizar arboles de núcleo, donde se almacena solo un árbol de expansión por grupo, que tiene como raíz el nodo mas cercano al centro. Entonces para hacer multidifusión se envía el paquete a la raíz y este se encarga de hacer la difusión.

5.3 ALGORITMOS DE CONTROL DE CONGESTIÓN

El control de congestión previene que se envíen mas paquetes de los que la red puede manejar de manera eficiente, o en caso de que suceda, trata de mitigar el daño. La diferencia con el control de flujo es que este se encarga de que un emisor no sature al receptor (punto a punto), mientras que el control de congestión se encarga de que los host (multi punto) no saturen la red.

Los algoritmos de control de flujo se clasifican de la siguiente manera:

- Ciclo abierto: se combate el problema mediante un buen diseño. Las decisiones que se toman son independientes del estado de la red. Se dividen en algoritmos que actúan en el origen y algoritmos que actúan en el destino. Ejemplos de estas politicas son:
    - Capa de enlace
        - Retransmisión selectiva vs retroceso n.
        - Tiempo de expiración de paquetes
        - Política de confirmación. Piggybacking
    - Capa de red
        - Algoritmo de enrutamiento
        - Tiempo de expiración de paquetes
        - Colas de entrada y salida en los routers
        - Orden en que se procesan los paquetes. Política de descarte
    - Capa de transporte (igual a la capa de enlace)
- Ciclo cerrado: se basan en un ciclo de retroalimentación. Tiene tres partes, primero se monitorea el sistema para detectar las congestiones, después se pasa la información a los lugares donde se pueda llevar a cabo acciones y por ultimo se ajusta la operación para solucionar el problema
    - De retroalimentación explicita: se regresan paquetes del punto de congestión al origen para avisar del problema.
    - De retroalimentación implícita: el origen deduce la existencia de una congestion haciendo observaciones locales.

Control de congestión en subredes de circuitos virtuales

- Control de admisión: el algoritmo consiste en dejar de aceptar solicitudes de conexión (creación de un nuevo circuito virtual) si la linea esta congestionada.
- Ruta alternativa: Se acepta la solicitud pero el circuito virtual se hace pasar por otro camino que no este congestionado
- Reserva de recursos: Se reservan los recursos antes de tiempo. Así se evita la congestión,

pero se desperdicia recursos.

Control de congestión en subredes de datagramas

En general lo que se hace es medir la utilización de las lineas. Cuando se observa que una linea esta alcanzando su limite, se puede elegir entre varias estrategias:

- Bit de advertencia: se activa un bit de advertencia en la cabecera de los paquetes que transitan por la linea congestionada. Cuando estos llegan a destino, el router activa el bit en la confirmación de recepción. De esta forma el router de origen se entera de la congestión y disminuye el trafico de acuerdo a la proporción de confirmaciones que reciba con el bit activado.
- Paquetes reguladores: El router envia un paquete avisando al origen que debe disminuir el trafico en un porcentaje x. El origen al recibir este paquete disminuye el envío, ignora todos los paquetes reguladores que reciba en cierto tiempo. Al finalizar el tiempo comienza a incrementar el trafico, a no ser que reciba otro paquete regulador.
- Paquetes reguladores salto a salto: igual al anterior, solo que obliga a disminuir el trafico generado a los routers intermedios por los que va pasando. De esta forma el destino ve aliviada su linea de forma mas rápida, ya que los buffers de los routers intermedios van soportando la carga.

Desprendimiento de carga

Consiste en eliminar paquetes cuando se alcanza un limite de cogestionamiento. Se pueden eliminar paquetes al azar, o utilizando alguna política inteligente (borrando los mas viejos primero, los mas nuevos, de acuerdo al contenido, etc). También se pueden agregar etiquetas a los paquetes indicando el nivel de importancia, así se borran primero los menos valiosos.

Otra alternativa es la deteccion temprana aleatoria. Consiste en eliminar paquetes aleatorios antes que la linea se congestione, cuando alcance cierto umbral de carga. En algunos protocolos (TCP) cuando se pierde un paquete se supone que es por sobrecarga de la linea, y se responde disminuyendo el trafico generado, de esta forma este algoritmo consigue su objetivo.

Control de fluctuación

Consiste en disminuir la varianza del retardo de los paquetes. Es decir, conseguir que todos tengan aproximadamente el mismo retardo. Para ello se puede colocar el tiempo esperado para cada salto en el paquete, entonces cada router puede retrasarlo cierto tiempo si el paquete va demasiado adelantado o sacarlo rápidamente si esta retrasado. El algoritmo que decide que paquete enviar primero puede utilizar esta información para enviar primero los mas retrasados.

5.4 CALIDAD DEL SERVICIO

Requerimientos

Un flujo es un conjunto de paquetes que van de un origen a un destino. La necesidad de estos se puede caracterizar por cuatro parámetros que determinan la calidad del servicio (QoS):

- Ancho de banda
- Confiabilidad

- Fluctuaciones
- Retardo

Las redes ATM clasifican los flujos en 4 tipos:

- Tasa de bits constantes (Telefonía)
- Tasa de bits variable en tiempo real (Videoconferencia comprimida)
- Tasa de bits variable no constante (Ver película por internet)
- Tasa de bits disponible (Transferencia de archivos)

Tecnicas para alcanzar la buena calidad del servicio

- Sobre aprovisionamiento: Consiste en proporcionar de suficiente capacidad de enrutadores, ancho de banda y bufferes a la red para que los paquetes fluyan con facilidad
- Almacenamiento en buffer: Lo paquetes que van llegando se almacenan en un buffer en el receptor antes de ser entregados. De esta forma disminuye las fluctuaciones, pero aumenta el retardo.
- Modelado de trafico: consiste en regular la tasa promedio y las ráfagas de la transmisión de datos en el servidor. Cuando se establece una nueva conexión, el cliente y la subred acuerdan cierto patrón de trafico. De esta forma la subred puede asegurarse que puede manejar el trafico.
- Algoritmos de cubeta con goteo: Es un algoritmo que se implementa en los host y se puede utiliza para que estos generen un flujo uniforme de paquetes. Consiste en utilizar una cola como buffer en el emisor y enviando paquetes de esta cola a la tasa deseada.
- Algoritmo de cubeta con tokens: Es similar al anterior, pero permite que se envíen ráfagas de paquetes rápidas durante cierto tiempo. Se puede usar en conjunto con el algoritmo de cubeta con goteo.
- Reservación de recursos: consiste en utilizar circuitos virtuales, reservando los recursos a utilizar (ancho de banda, cpu y buffers) de antemano para garantizar la calidad del servicio.
- Control de admisión: Esta relacionada con la reservación de recursos. Cuando se desea crear un circuito nuevo, el origen envía una **especificación de flujo** por el circuito. Esta indica datos como la tasa de la cubeta con tokens, la tasa pico de datos, los tamaños máximos y mínimos de los paquetes, etc. Cada router que esta en el circuito revisa esta especificación y la modifica en caso de no poder garantizarla. Cuando llega al otro lado se puede establecer los parámetros.
- Enrutamiento proporcional: se divide el trafico de un flujo por varios rutas. Para definir la cantidad de paquetes que se envía por cada ruta se puede utilizar información local, como el ancho de banda de la linea.
- Estandarización de paquetes: controla el orden en que se envían los paquetes en un router. En vez de enviar los paquetes en el orden de llegada, se utilizan varias colas de llegada, y se hace un Round-Robin sobre estas para enviar los paquetes. Esto se conoce como **encolamiento justo**. Para mejorarlo, en vez de hacer un Round-Robin, se puede ordenar los paquetes que están al frente de cada cola de acuerdo a su tamaño, enviando primero los mas pequeños. También se le puede agregar un peso a cada cola. El nombre del algoritmo que implementa esta modificación es **encolamiento justo ponderado**.

5.6 LA CAPA DE RED DE INTERNET

Falta terminar...

Resumen 2er parcial de Redes y Comunicación de Datos II

6.4 LOS PROTOCOLOS DE TRANSPORTE DE INTERNET: UDP

UDP

UDP (Protocolo de datagramas de Usuario) es un protocolo no orientado a la conexión. Esta compuesto por un encabezado de 8 bytes seguido por la carga útil. El encabezado tiene 4 partes de 2 bytes:

- Puerto de origen
- Puerto destino
- Longitud del UDP (incluyendo los 8 bytes del encabezado)
- Suma de verificación opcional. Se puede ignorar colocando el valor '0'

Un área donde UDP es especialmente útil es en las situaciones cliente-servidor, DNS por ejemplo.

Llamada a procedimiento remoto

Otra área donde se utiliza UDP es en RCP (llamada a procedimiento remoto). Este consiste en hacer que una llamada a un procedimiento remoto sea lo mas parecida a una llamada local. Para funcionar, el procedimiento en la maquina cliente se debe enlazar con un pequeño procedimiento de biblioteca llamado **stub del cliente** y el servidor con un procedimiento llamado **stub del servidor**. Entonces, el cliente puede llamar a un procedimiento del stub del cliente, este se encarga de empaquetar los parámetros de la llamada en un mensaje **(marshaling)** y hacer la llamada correspondiente al sistema para enviar el mensaje al servidor. El kernel del servidor entrega el mensaje al stub del servidor, este llama al procedimiento correspondiente y luego se realiza el camino inverso para devolver la respuesta. Un claro problema con el RCP es que no puede trabajar con punteros, ya que los espacios de memoria son distintos. Las variable globales y las funciones con numero y tipo de parámetros variable (printf) también son un problema.

Protocolo de transporte en tiempo real

El protocolo RTP (real time protocol) se utiliza en aplicaciones multimedia en tiempo real. Este protocolo se ubica en el espacio de aplicación y normalmente trabaja sobre UDP. La función básica que tiene es la de multiplexar varios flujos de datos en tiempo real en un solo flujo de mensajes UDP. Puede ser unidifusión o multidifusion.

Cada paquete enviado cuenta con un numero incremental y una marcar temporal. El primero sirve para que el receptor se de cuenta cuando falta un paquete y de alguna forma solucione el problema (interpolando por ejemplo). El segundo sirve para reducir las fluctuaciones en el receptor al independizar la reproducción del orden de llegada de los paquetes y, ademas, para sincronizar varios flujos.

En el encabezado además se establece el formato de codificación de los datos (PCM, MP3, etc. en el caso de audio). Al encontrarse esta información en cada paquete el formato de codificación puede cambiar en el medio de la transmisión.

Existe otro protocolo llamado RTCP (Real Time Control Protocol) que sirve para sincronizar y retroalimentar a RCP, pero no enviá datos. Es decir, con este protocolo se puede avisar de posibles congestiones, retardos, fluctuaciones, etc. De esta forma el proceso de codificador puede adaptar la tasa de datos enviados al estado de la red. Además, RTCP sirve para darle nombres a los orígenes de datos, por ejemplo, para desplegarlos en la pantalla del usuario.

6.5 LOS PROTOCOLOS DE TRANSPORTE DE INTERNET: TCP

Introducción a TCP

TCP se diseño específicamente para proporcionar un flujo de bytes confiables de extremo a extremo a través de una interred no confiable.

Cada maquina que soporta TCP tiene una entidad de transporte que puede ser un proceso de usuario, una biblioteca o parte del kernel. Esta se encarga de convertir los flujos de datos del usuario en fragmentos de hasta 64kb (normalmente 1460 bytes) y mandarlos como datagramas IP independientes. Además se encarga de terminar los temporizadores de reenvió y reconstruir el flujo en orden a medida que llegan paquetes IP, garantizando de esta forma la confiabilidad que IP no proporciona.

El modelo del servicio TCP

Todas las conexiones TCP son duplex total y de punto a punto. No soporta la multidifusión ni la difusión. TCP es un flujo de bytes, no de mensajes, por lo que los limites de los mensajes no se preservan extremo a extremo.

Cuando una aplicación le pasa datos a TCP, este decide si los manda inmediatamente o los almacena en buffer para enviarlos cuando tenga mas. Para evitar esto se debe utilizar el indicador PUSH, que es una señal para TCP de que no debe retrasar la transmisión. El efecto es que ni la aplicación que esta enviando ni la que recibe utilizan buffers.

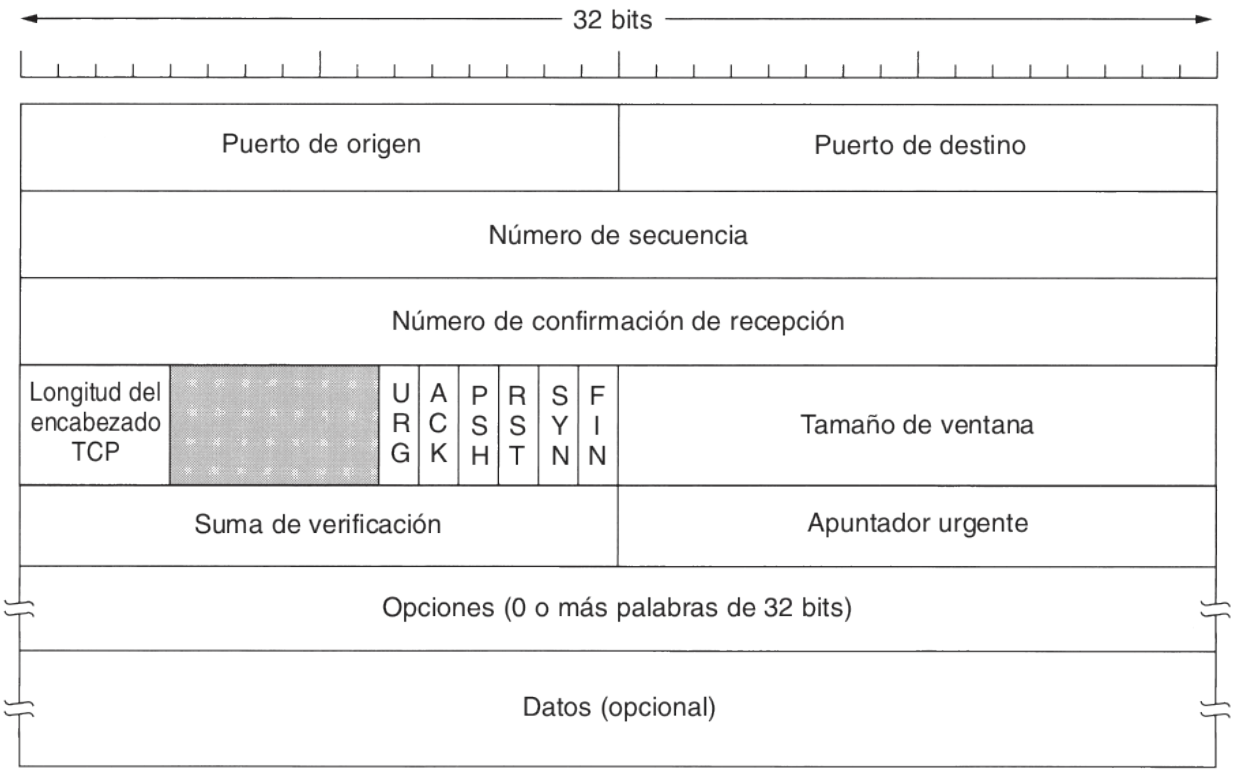
Otra característica son los datos URGENT. Este es un puntero que se encuentra en el encabezado que marca los datos urgentes que contiene el paquete (marca el ultimo, el principio lo debe suponer la aplicación que recibe el paquete).

El protocolo TCP

Cada byte en una conexión TCP tiene su propio numero de secuencia de 32 bytes. Esto es un limitante ya que este numero se utiliza para confirmaciones de recepción y para llevar adelante el mecanismo de la ventana. Las conexiones hoy en día son muy rápidas y consumen todos los números de secuencia en poco tiempo. Este problema es muy común en las comunicaciones satélites.

TCP decide el tamaño de los segmentos que enviá. Hay dos limites que restringen el tamaño de estos paquetes, el tamaño de paquete IP (65515 bytes) y el unidad maxima de transferencia(MTU) de la red, que en ethernet es de 1500 bytes.

El encabezado del segmento TCP



- Puertos de origen y destino: identifican los puntos terminales de la conexión.
- Numero de secuencia y numero de recepción: desempeñan sus funciones normales. Cada byte de datos esta numerado. El segundo identifica el próximo byte esperado, no el ultimo recibido.
- Longitud del encabezado: Cantidad de palabras de 32 bits que ocupa el encabezado.
- URG: indica si esta en uso el apuntador de urgente.
- ACK: indica que se debe considerar el numero de confirmación de recepción.
- PSH (push): indica que los datos se pasen inmediatamente a la aplicación, sin almacenarse previamente en buffer.
- RST (reset): sirve para restablecer una conexión. También se utiliza para rechazar un segmento con errores o una solicitud de conexión.
- SYN: se utiliza para establecer una conexión.
- FIN: se utiliza para cerrar una conexión. Indica que el emisor no tiene mas datos para enviar.
- Tamaño de la ventana: indica cuantos bytes pueden enviarse comenzando desde el byte cuya recepción se confirmo. Se puede enviar un 0 para parar el envió de datos.
- Suma de verificación: es una suma del encabezado, los datos y un pseudo-encabezado formado por las direcciones IP de origen y destino, el numero de protocolo (6) y la cuenta de bytes del segmento TCP (incluido el encabezado). La inclusión del pseudo-encabezado viola la jerarquía de capas, pero ayuda a detectar paquetes mal formados. UDP también lo utiliza.
- Opciones: permite agregar características no cubiertas por el encabezado. Se lo utiliza por ejemplo para informar del tamaño máximo de segmentos con el que puede trabajar un host.

También para negociar una factor de escala para la ventana (permite utilizar tamaños de ventanas de hasta 30 bits), para evitar el problema de las conexiones de gran ancho de banda y alto retardo. Una ultima aplicación del campo de opciones es para realizar rechazo selectivo, en lugar de retroceso-n.

Establecimiento de una conexión TCP

Para establecer una conexión se utiliza el acuerdo de 3 vías. En primer lugar el servidor debe ejecutar la primitiva LISTEN y ACCEPT para esperar pasivamente una conexión. Del lado cliente se debe ejecutar la primitiva CONNECT con la dirección del servidor para iniciar el proceso. En el caso de que dos host intentan establecer una conexión el resultado es que solo se acepta una ya que el identificador de la conexión (puntos terminales) es el mismo, solo se agrega una entrada a la tabla. Los números de secuencia utilizados se basan en un esquema de reloj con ciclo 4 micro segundos, además, se espera 120 micro segundos para reiniciar una conexión cuando se cae un host.

Liberación de una conexión TCP

Cada lado de la conexión se libera de forma independiente. Cualquiera de las dos partes puede enviar un segmento con la bandera FIN para indicar que no tiene mas datos para enviar. Al confirmarse la recepción ese lado se apaga pero puede continuar el otro lado enviando datos.

Para evitar el problema de los dos ejércitos se usan temporizadores. Si no llega una respuesta a un FIN en dos tiempo de vida de paquete, el emisor del FIN libera la conexión.

Política de transmisión del TCP

La administración de ventanas en TCP no esta vinculada a la confirmación de recepción como en otros protocolos de enlace de datos. En TCP el receptor le indica cual es el tamaño de ventana libre que le queda al emisor para poder enviar. Puede enviarle un 0 para detener el envío de datos, en cuyo caso el emisor solo puede enviar datos urgentes o segmentos de 1 byte para solicitar que el emisor reanuncie el siguiente byte esperado y el tamaño de ventada disponible. Este ultimo mecanismo se utiliza para evitar bloqueos irreversibles al perderse un anuncio de ventana.

No se requiere que los emisores envíen los datos tan pronto reciba desde la aplicación, ni que los receptores se los pasen a la aplicación y envíen confirmación tan pronto estos lleguen. Estas libertades pueden explotarse para mejorar el desempeño del protocolo, por ejemplo:

- Retardando las confirmaciones de recepción y las actualizaciones de ventana 500 μseg con la esperanza que aparezcan datos a los cuales acoplarse.
- Algoritmo de Nagle: al llegar al emisor datos 1 byte a la vez, se enviá el primer dato y se almacenan en buffer el resto hasta que llegue la confirmación de recepción del primero. Luego se envían todos juntos los caracteres almacenados en buffer y se vuelven a almacenar los que llegan de la aplicación. El algoritmo enviá el segmento si han entrado suficientes datos para llenar la mitad de la ventana o la totalidad de un segmento.
- Solución de Clark: consiste en evitar que el emisor envié la actualización de la ventana hasta que no tenga la mitad del buffer vació o hasta que no pueda manejar el tamaño de segmento máximo declarado, lo que sea mas pequeño. Esto evita el problema de la ventana tonta, que ocurre cuando la aplicación del lado receptor lee de a un byte a la vez.
- El receptor puede bloquear la primitiva READ hasta que no halla suficientes datos en buffer. De esta forma se evita la sobrecarga en la transferencia de archivos por ejemplo.

- Otra política es almacenar los segmentos que lleguen fuera de orden. De esta forma se pueden aprovechar cuando sean necesarios.

Control de congestión en TCP

Para controlar la congestión, el primer paso es su detección. Esto se realiza mirando la expiración del temporizador. En otra época esto podía suceder por ruido en la linea, pero actualmente, con las lineas de fibra, solamente ocurre por congestión en la red.

Para evitar la congestión, los emisores tienen dos ventanas, la ventana del receptor y la de congestión de la red. La cantidad de bytes que pueden enviarse es la cifra menor de las dos ventanas. Al establecer una conexión, el emisor le asigna a la ventana de congestión el tamaña de segmento máximo usado por la conexión. Mientras los segmentos se confirmen y no se alcance el tamaño de la ventana receptora, el tamaño de esta ventana se duplica. Si se expira un temporizador, la ventana se divide a la mitad y el algoritmo se detiene. Esto se conoce como **arranque lento**.

El algoritmo de control de congestión de Internet utiliza un tercer parámetro, el **umbral**, inicialmente de 64 KB, además de las ventanas de recepción y congestión. Al ocurrir una expiración del temporizador, se establece el umbral en la mitad de la ventana de congestión actual, y la ventana de congestión se restablece a un segmento máximo. Luego se usa el arranque lento para determinar lo que puede manejar la red, excepto que el crecimiento exponencial termina al alcanzar el umbral. A partir de este punto, las transmisiones exitosas aumentan linealmente la ventana de congestión (en un segmento máximo por ráfaga) . Si llega un paquete SOURCE QUENCH de ICMP y pasa al TCP, este evento será tratado de la misma manera que una expiración del temporizador.

Administración de temporizadores del TCP

TCP utiliza varios temporizadores:

- De retransmisión: solicita una retransmisión de segmento.
- De persistencia: sirve para evitar el bloqueo irreversible que se da cuando se pierde una actualización de ventana. Obliga al emisor a solicitar una actualización si no llego ninguna al expirar el temporizador.
- De seguir con vida: cuando una conexión estuvo inactiva durante cierto tiempo y este temporizador termina, se comprueba que la otra parte todavía este conectada. Si no se recibe respuesta la conexión finaliza, sino se reinicia el temporizador.
- El tiempo de espera: garantiza que todos los paquetes de una conexión expiren al cerrar la conexión. Dura el doble del tiempo máximo de vida de los segmentos.

Para calcular el tiempo de expiración del temporizador de retransmisión se utiliza un algoritmo muy dinámico que continuamente esta actualizando el RTT (round-trip time) con la siguiente formula:

$$RTT = \alpha\, RTT + (1-\alpha)\, M$$

Donde M es el tiempo que demora un segmento en viajar hasta el receptor mas el tiempo que demora la confirmación en volver. Ademas se mide la variación media de tiempos:

$$D = \alpha\, D + (1-\alpha)\,|RTT - M|$$

Entonces el tiempo de expiración T es igual a:

$$T = RTT + 4\, D$$

La actualización de RTT y D no se debe realizar con paquetes retransmitidos, ya que no se sabe si la

confirmación es del primer envío o de la retransmisión. En estos casos Karn propone duplicar RTT y D.

7.1 DNS - EL SISTEMA DE NOMBRES DE DOMINIO

El espacio de nombres del DNS

Internet se divide en dominios de nivel superior, estos se dividen en subdominios, los cuales, a su vez, también se dividen, y así sucesivamente. Un dominio hoja puede contener un solo host, o puede representar a una compañía y contener miles de hosts.

Los dominios de nivel superior se dividen en dos categorías: genéricos (com, edu, gov, etc.) y de país (ar, br, ch, etc.).

En general, obtener un dominio de segundo nivel, como nombre-de-compañía.com, es fácil. Simplemente se necesita ir con el registrador del dominio de nivel superior correspondiente (com, en este caso) para ver si el nombre deseado está disponible y si no es la marca registrada de alguien más. Si no hay problemas, el solicitante paga una pequeña cuota anual y obtiene el nombre.

Cada dominio se nombra por la ruta hacia arriba desde él a la raíz (sin nombre). Los componentes se separan con puntos. Los nombres de dominio pueden ser absolutos o relativos. Un nombre de dominio absoluto termina con un punto (por ejemplo, eng.sun.com.), y uno relativo no. Los nombres relativos tienen que interpretarse en algún contexto para determinar de manera única su significado verdadero. En ambos casos, un dominio nombrado hace referencia a un nodo específico del árbol y a todos los nodos por debajo de él.

Los nombres de dominio no hacen distinción entre mayúsculas y minúsculas. Los nombres de componentes pueden ser de hasta 63 caracteres de longitud, y los de ruta completa de hasta 255 caracteres.

Casi todas las organizaciones de Estados Unidos están bajo un dominio genérico, y casi todas las de fuera de Estados Unidos están bajo el dominio de su país. No hay ninguna regla que impida registrarse bajo dos dominios .

Los nombres reflejan los límites organizacionales, no las redes físicas.

Registros de recursos

Cada dominio, sea un host individual o un dominio de nivel superior, puede tener un grupo de registros de recursos asociados a él. Cuando un resolvedor da un nombre de dominio al DNS, lo que recibe son los registros de recursos asociados a ese nombre.

Un registro de recursos tiene cinco tuplas. Aunque éstas se codifican en binario por cuestión de eficiencia, en la mayoría de las presentaciones, los registros de recursos se dan como texto ASCII, una línea por registro de recursos. El formato que usaremos es el siguiente:

Nombre_dominio Tiempo_de_vida Clase Tipo Valor

- El Nombre_dominio indica el dominio al que pertenece este registro. Es decir, es la clave de búsqueda utilizada para atender las consultas
- El campo de Tiempo_de_vida es una indicación de la estabilidad del registro.
- El tercer campo de cada registro de recursos es Class. Para la información de Internet, siempre es IN. Para información que no es de Internet, se pueden utilizar otros códigos, pero en la práctica, éstos raramente se ven.

- El campo Tipo indica el tipo de registro de que se trata, puede ser:
  - SOA: proporciona el nombre de la fuente primaria de información sobre la zona del servidor de nombres (que se describe más adelante), la dirección de correo electrónico de su administrador, un número de serie único y varias banderas y temporizadores.
  - A: dirección IP del host. Cada host puede tener una o mas direcciones asociadas. DNS se puede configurar para iterar a través de éstas.
  - MX: especifica el nombre del dominio que está preparado para aceptar correo electrónico del dominio especificado.
  - NS: especifican servidores de nombres.
  - CNAME: permiten la creación de alias.
  - PTR: apunta a otro nombre, es un tipo de datos DNS normal, cuya interpretación depende del contexto en el que se encontró . Normalmente se utiliza para hacer búsquedas invertidas.
  - HINFO: permite que la gente conozca el tipo de máquina y sistema operativo al que corresponde un dominio.
  - TXT: permite a los dominios identificarse de modos arbitrarios.
- Por último, llegamos al campo Valor. Este campo puede ser un número, un nombre de dominio o una cadena ASCII.

Servidores de nombres

el espacio de nombres DNS se divide en zonas no traslapantes. Cada zona contiene una parte del árbol y también contiene servidores de nombres que tienen la información de autorización correspondiente a esa zona. Por lo general, una zona tendrá un servidor de nombres primario, que obtiene su información de un archivo en su disco, y uno o más servidores de nombres secundarios, que obtienen su información del primario.

Cuando un resolvedor tiene una consulta referente a un nombre de dominio, la pasa a uno de los servidores de nombres locales. Si el dominio que se busca cae bajo la jurisdicción del servidor de nombres, devuelve los registros de **recursos autorizados**. Un registro autorizado es uno que proviene de la autoridad que administra el registro y, por lo tanto, siempre es correcto. Los registros autorizados contrastan con los registros en caché, que podrían no estar actualizados. Por otro lado, si el dominio es remoto y no hay información disponible localmente sobre el dominio solicitado, el servidor de nombres envía un mensaje de consulta al servidor de nombres de nivel superior en el que le solicita dicho dominio. Puesto que cada solicitud es de un cliente a un servidor, el registro de recursos solicitado regresa por el mismo camino.

Una vez que estos registros regresan al primer servidor de nombres, se almacenan en caché, por si se necesitan posteriormente. Sin embargo, esta información no es autorizada , por este motivo las entradas de caché no deben vivir demasiado tiempo. Ésta es la razón por la cual el campo Tiempo_de_vida se incluye en cada registro de recursos; indica a los servidores de nombres remotos cuánto tiempo deben mantener en caché los registros.

Este procedimiento se conoce como **consulta recursiva.** Es posible un procedimiento alternativo. En él, cuando una consulta no puede satisfacerse localmente, ésta falla, pero se devuelve el nombre del siguiente servidor a intentar a lo largo de la línea.

También vale la pena indicar que cuando un cliente DNS no recibe una respuesta antes de que

termine su temporizador, por lo general probará con otro servidor la siguiente vez.

7.2 CORREO ELECTRÓNICO

Arquitectura y servicios

Consisten en dos subsistemas:

- Los agentes de usuario: permiten a la gente leer y enviar correo electrónico. Son programas locales que proporcionan un método basado en comandos o en una interfaz gráfica para interactuar con el sistema de correo electrónico.
- Los agentes de transferencia de mensajes: mueven los mensajes del origen al destino. Por lo común son demonios del sistema que operan en segundo plano y mueven correo electrónico a través del sistema.

Los sistemas de correo electrónico generalmente desempeñan cinco funciones básicas:

- La redacción: se refiere al proceso de crear mensajes y respuestas.
- La transferencia: se refiere a mover mensajes del remitente al destinatario.
- La generación del informe tiene que ver con indicar al remitente lo que ocurrió con el mensaje.
- La visualización de los mensajes de entrada es necesaria para que la gente pueda leer su correo electrónico.
- La disposición es el paso final y tiene que ver con lo que el destinatario hace con el mensaje una vez que lo recibe.

Otras características avanzadas que puede presentar el correo electronico son: la creación de buzones de correo, las listas de correo, copias ocultas, correo de alta prioridad, correo encriptado, destinatario alterno, etc.

El correo electrónico se divide en el sobre y su contenido. El sobre encapsula el mensaje; contiene toda la información necesaria para transportar el mensaje. Los agentes de transporte del mensaje usan el sobre para enrutar. El mensaje dentro del sobre contiene dos partes: el encabezado y el cuerpo. El encabezado contiene información de control para los agentes de usuario. El cuerpo es por completo para el destinatario humano.

El agente de usuario

Un agente de usuario normalmente es un programa (a veces llamado lector de correo) que acepta una variedad de comandos para redactar, recibir y contestar los mensajes, así como para manipular los buzones de correo.

Envió de correo: El usuario debe proporcionar el mensaje, la dirección de destino y, posiblemente, algunos otros parámetros. El mensaje puede producirse con un editor de texto independiente o, posiblemente, un editor de texto incorporado en el agente de usuario. La dirección de destino debe estar en un formato que el agente de usuario pueda manejar. Muchos agentes de usuario esperan direcciones DNS de la forma usuario@dirección-dns. Existen otros formatos como el propuesto por X.400. Estas se componen de pares atributo = valor, separadas por diagonales, por ejemplo,

/C=US/ST=MASSACHUSETTS/L=CAMBRIDGE/PA=360 MEMORIAL DR./CN=KEN SMITH/

Esta dirección especifica un país, estado, localidad, dirección personal y nombre común . Son

posibles muchos otros atributos, de modo que puede enviarse correo electrónico a alguien cuyo nombre no se conoce, siempre y cuando se conozca un número suficiente de atributos.

Lectura de correo: Por lo común, cuando se inicia un agente de usuario, buscará en el buzón del usuario el correo electrónico recibido, antes de presentar otra cosa en la pantalla. Después puede anunciar la cantidad de mensajes en el buzón o presentar un resumen de una línea de cada uno y esperar un comando.

Formatos de mensaje

RFC 822

Los mensajes consisten en un sobre primitivo (descrito en el RFC 821), algunos campos de encabezado, una línea en blanco y el cuerpo del mensaje. Cada campo de encabezado consiste en una sola línea de texto ASCII que contiene el nombre del campo, dos puntos (:) y, para la mayoría de los campos, un valor. El RFC 822 es un estándar viejo y no distingue los campos del sobre de los de encabezado. Aunque se revisó en el RFC 2822, no fue posible restaurarlo por completo debido a su uso amplio. En el uso normal, el agente de usuario construye un mensaje y lo pasa al agente de transferencia de mensajes, quien después usa algunos campos del encabezado para construir el sobre . Principales campos de encabezado relacionados con el transporte del mensaje:

Encabezado	Significado
To:	Direcciones de correo electrónico de los destinatarios primarios
Cc:	Direcciones de correo electrónico de los destinatarios secundarios
Bcc:	Direcciones de correo electrónico para las copias ocultas
From:	Persona o personas que crearon el mensaje
Sender:	Dirección de correo electrónico del remitente
Received:	Línea agregada por cada agente de transferencia en la ruta
Return-Path:	Puede usarse para identificar una ruta de regreso al remitente

Otros campos de encabezado:

Encabezado	Significado
Date:	Fecha y hora de envío del mensaje
Reply-To:	Dirección de corre electrónico a la que deben enviarse las contestaciones
Message-Id:	Número único para referencia posterior a este mensaje
In-Reply-To:	Identificador del mensaje al que éste responde
References:	Otros identificadores de mensaje pertinentes
Keywords:	Claves seleccionadas por el usuario
Subject:	Resumen corto del mensaje para desplegar en una línea

MIME—Extensiones Multipropósito de Correo Internet

Se propuso en el RFC 1341 y se actualizó en los RFCs 2045-2049 . La idea básica de MIME es

continuar usando el formato RFC 822, pero agregar una estructura al cuerpo del mensaje y definir reglas de codificación para los mensajes no ASCII .

MIME define cinco nuevos encabezados de mensaje :

Encabezado	Significado
MIME-Version:	Identifica la versión de MIME
Content-Description:	Cadena de texto que describe el contenido
Content-Id:	Identificador único
Content-Transfer-Encoding:	Cómo se envuelve el mensaje para su transmisión
Content-Type:	Naturaleza del mensaje

Content-Transfer-Encoding indica la manera en que está envuelto el cuerpo para su transmisión a través de una red donde se podría tener problemas con los caracteres distintos de las letras, números y signos de puntuación. Se proporcionan cinco esquemas :

- Texto ASCII de 7 bits: puede transportarse sin problemas. Las lineas deben tener menos de 1000 caracteres
- Texto ASCII de 8 bits: viola el protocolo de Internet, pero se utiliza en algunas partes del mundo. También respeta el limite de 1000 caracteres
- Codificación binaria: los ejecutables por ejemplo. No se da ninguna garantía de que lleguen, pero sin embargo se usa. No respetan el limite de 1000 caracteres
- Codificación base64 o armadura ASCII: En este esquema se dividen grupos de 24 bits en unidades de 6 bits, y cada unidad se envía como un carácter ASCII legal . Los retornos de carro y avances de línea se ignoran, por lo que puede introducirse a voluntad para mantener la línea lo suficientemente corta. Puede enviarse un texto binario arbitrario usando este esquema.
- Codificación entrecomillada imprimible : se utiliza para mensajes que son casi completamente ASCII. Simplemente es ASCII de 7 bits, con todos los caracteres por encima de 127 codificados como un signo de igual seguido del valor del carácter en dos dígitos hexadecimales.
- Cuando hay razones válidas para no usar uno de estos esquemas, es posible especificar una codificación definida por el usuario en el encabezado Content-Transfer-Encoding .

Content-Type especifica la naturaleza del cuerpo del mensaje. En el RFC 2045 hay siete tipos definidos, cada uno de los cuales tiene uno o más subtipos. El tipo y el subtipo se separan mediante una diagonal . El subtipo debe indicarse de manera explícita en el encabezado; no se proporcionan valores predeterminados. La lista inicial de tipos y subtipos esta especificada en el RFC 2045. Se han agregado muchos otros desde entonces, y se agregan entradas nuevas todo el tiempo, a medida que surge la necesidad. Los tipos y subtipos definidos en el RFC 2045 son:

Tipo	Subtipo	Descripción
Texto	Plano	Texto sin formato
	Enriquecido	Texto con comandos de formato sencillos
Imagen	Gif	Imagen fija en formato GIF
	Jpeg	Imagen fija en formato JPEG
Audio	Básico	Sonido
Vídeo	Mpeg	Película en formato MPEG
Aplicación	Octet-stream	Secuencia de bytes no interpretada
	Postscript	Documento imprimible en PostScript
Mensaje	Rfc822	Mensaje MIME RFC 822
	Parcial	Mensaje dividido para su transmisión
	Externo	El mensaje mismo debe obtenerse de la red
Multipartes	Mezclado	Partes independientes en el orden especificado
	Alternativa	Mismo mensaje en diferentes formatos
	Paralelo	Las partes deben verse en forma simultánea
	Compendio	Cada parte es un mensaje RFC 822 completo

Transferencia de mensajes

SMTP—Protocolo Simple de Transporte de Correo

En Internet, el correo electrónico se entrega al hacer que la máquina de origen establezca una conexión TCP con el puerto 25 de la máquina de destino. Escuchando en este puerto está un demonio de correo electrónico que habla con el SMTP. Este demonio acepta conexiones de entrada y copia mensajes de ellas a los buzones adecuados. Si no puede entregarse un mensaje, se devuelve al remitente un informe de error que contiene la primera parte del mensaje que no pudo entregarse.

SMTP es un protocolo ASCII sencillo. Después de establecer la conexión TCP con el puerto 25, el servidor comienza enviando una línea de texto que proporciona su identidad e indica si está preparado o no para recibir correo. Si no lo está, el cliente libera la conexión y lo intenta después. Si lo esta, el cliente anuncia de quién proviene el mensaje, y a quién está dirigido. Si existe el destinatario en el destino, el servidor da al cliente permiso para enviar el mensaje. A continuación el cliente envía el mensaje y el servidor confirma su recepción. Por lo general, no se requieren sumas de verificación porque TCP proporciona un flujo de bytes confiable. Si hay más correo electrónico, se envía ahora. Una vez que todo el correo electrónico ha sido intercambiado en ambas direcciones, se libera la conexión.

Aunque el protocolo SMTP está bien definido, pueden surgir algunos problemas:

- Longitud del mensaje: Algunas implementaciones más viejas no pueden manejar mensajes mayores que 64 KB.

- Terminaciones de temporizador: Si el cliente y el servidor tienen temporizadores diferentes, uno de ellos puede terminar mientras que el otro continúa trabajando
- Tormentas de correo infinitas: si el host 1 contiene la lista de correo A y el host 2 contiene la lista de correo B y cada lista contiene una entrada para la otra lista, entonces un mensaje enviado a cualquiera de las listas generará una cantidad sin fin de tráfico de correo.

Para superar algunos de estos problemas, se ha definido el SMTP extendido (ESMTP) en el RFC 2821. Los clientes que deseen usarlo deben enviar inicialmente un mensaje EHLO, en lugar de HELO. Si el saludo se rechaza, esto indica que el servidor es un servidor SMTP normal, y el cliente debe proceder de la manera normal. Si se acepta el EHLO, entonces se permiten los comandos y parámetros nuevos.

Entrega final

Con el advenimiento de personas que acceden a Internet llamando a su ISP por medio de un módem, es decir, que no están todo el tiempo en linea surge un problema para el envió de correo cuando el receptor no esta en linea. Para solucionar esto se hace que un agente de transferencia de mensajes en una máquina ISP acepte correo electrónico para sus clientes y lo almacene en sus buzones en una máquina ISP . Luego el receptor accede a este buzon utilizando alguno de los siguientes protocolos

POP3

POP3 inicia cuando el usuario arranca el lector de correo. Éste llama al ISP (a menos que ya haya una conexión) y establece una conexión TCP con el agente de transferencia de mensajes en el puerto 110. Una vez que se ha establecido la conexión, el protocolo POP3 pasa por tres estados

en secuencia:

1. Autorización: inicio de sesión por parte del usuario (se utilizan los comandos USER y PASS)
2. Transacción: el usuario descarga los correos y los marca para su eliminación posterior (LIST, RETR y DELE).
3. Actualización: se encarga de que los mensajes de correo electrónico se eliminen realmente (QUIT).

IMAP

IMAP supone que todo el correo electrónico permanecerá en el servidor de manera indefinida. Esta orientado a usuarios que lean su correo desde varias estaciones de trabajo. Proporciona mecanismos para crear, destruir y manipular múltiples buzones en el servidor. De esta forma, un usuario puede mantener un buzón para cada uno de sus contactos y colocar ahí mensajes de la bandeja de entrada después de que se han leído. También brinda mecanismos para leer mensajes o incluso partes de un mensaje, una característica útil cuando se utiliza un módem lento.

Otra característica del protocolo es la capacidad de dirigir correo no por número de llegada sino utilizando atributos (por ejemplo, dame el primer mensaje de Juan). A diferencia de POP3, IMAP también puede aceptar correo saliente para enviarlo al destino, así como entregar correo electrónico entrante. El estilo general del protocolo IMAP es similar al de POP3 excepto que hay docenas de comandos. A continuación una comparativa entre los dos protocolos

Característica	POP3	IMAP
En dónde se define el protocolo	RFC 1939	RFC 2060
Puerto TCP utilizado	110	143
En dónde se almacena el correo electrónico	PC del usuario	Servidor
En dónde se lee el correo electrónico	Sin conexión	En línea
Tiempo de conexión requerido	Poco	Mucho
Uso de recursos del servidor	Mínimo	Amplio
Múltiples buzones	No	Sí
Quién respalda los buzones	Usuario	ISP
Bueno para los usuarios móviles	No	Sí
Control del usuario sobre la descarga	Poco	Mucho
Descargas parciales de mensajes	No	Sí
¿Es un problema el espacio en disco?	No	Con el tiempo podría serlo
Sencillo de implementar	Sí	No
Soporte amplio	Sí	En crecimiento

Correo de Web

Este sistema utiliza una pagina Web para mostrar el correo a sus usuarios. El usuario primero debe colocar su nombre y contraseña. Esta pagina Web le permite al usuario leer, eliminar mensajes, etc.

Características de entrega

Mas allá del protocolo de entrega que se utilice, muchos sistemas proporcionan ganchos para procesamiento adicional de correo electrónico entrante . Por ejemplo:

- filtros para el correo entrante: los correos que entrar se colocan en un buzón u otro dependiendo de diversos datos, como el remitente, o el asunto.
- Filtro de correo basura: el ISP mira si el remitente es un spammer, o si el asunto de mensaje se repitió en muchos correos de sus clientes, y de esta forma puede suponer que el correo es basura, y lo coloca en un buzón aparte.
- Reenvió automático de correo a otra dirección.
- Demonio de vacaciones: contesta el correo entrante con un mensaje personalizado.

7.3 WORLD WIDE WEB

Panorama de la arquitectura

Desde el punto de vista del usuario, Web consiste en un enorme conjunto de documentos a nivel mundial, generalmente llamados páginas Web. Cada página puede contener vínculos (apuntadores) a otras páginas relacionadas en cualquier lugar del mundo. Los usuarios pueden seguir un vínculo haciendo clic en él, lo que los lleva a la página apuntada. Este proceso puede repetirse de manera indefinida.

Las páginas se ven mediante un programa llamado navegador. El navegador obtiene la página solicitada, interpreta el texto y los comandos de formateo que contienen, y despliega la página, adecuadamente formateada, en la pantalla.

El cliente

En esencia, un navegador es un programa que puede desplegar una página Web y atrapar los clics que se hacen en los elementos de la página desplegada. Cuando se selecciona un elemento, el navegador sigue el hipervínculo y obtiene la página seleccionada. Por lo tanto, el hipervínculo incrustado necesita una manera de nombrar cualquier página que se encuentre en Web. Las páginas se nombran utilizando URLs (Localizadores Uniformes de Recursos). Un URL tiene tres partes: el nombre del protocolo (http), el nombre DNS de la máquina donde se localiza la página (www.abcd.com) y (por lo general) el nombre del archivo que contiene la página (productos.html) .

Para poder desplegar la nueva página, el navegador tiene que entender su formato. Para permitir que todos los navegadores entiendan todas las páginas Web, éstas se escriben en un lenguaje estandarizado llamado HTML, el cual las describe.

No todas las páginas contienen HTML. Una página puede consistir en un documento con formato PDF, GIF, JPEG, MP3, etc. Para abrir estos archivos los navegadores utilizan plug-ins o aplicaciones auxiliares.

El servidor

Un servidor Web generalmente recibe solicitudes de paginas y responde con el archivo solicitado. El proceso general es el siguiente:

1. Resuelve el nombre de la página Web solicitada.
2. Autentica al cliente, de ser necesario.
3. Realiza control de acceso en el cliente.
4. Realiza control de acceso en la página Web.
5. Verifica el caché.
6. Obtiene del disco la página solicitada si esta no estaba en cache.
7. Determina el tipo MIME que se incluirá en la respuesta.
8. Se encarga de diversos detalles.
9. Regresa la respuesta al cliente.
10. Realiza una entrada en el registro del servidor.

Como se puede ver, los servidores siempre utilizan cache para evitar el proceso de leer cada pagina del disco.

Otras técnicas para que el servidor pueda responder mas solicitudes, es la utilización de varios subprocesos dentro de cada maquina y varios discos. Un subproceso recibe todos los pedidos y se los pasa a otro subproceso para que se encargue de responderlo. Todos estos comparten la misma memoria cache.

También se puede utilizar varias maquinas en lugar de varios subprocesos dentro de la misma maquina. Esto se conoce como **granja de servidores**. La desventaja es que la memoria cache no se comparte y que es necesaria una maquina intermediaria para responder las solicitudes y delegar los trabajos. El primer problema se intenta solucionar haciendo que cada maquina responda siempre a las mismas solicitudes de pagina, de esta forma se “especializa” en esas solicitudes. El segundo

problema se resuelve con **transferencia TCP**. Con ésta, el punto final de la conexión TCP se pasa al nodo de procesamiento a fin de que pueda contestar directamente al cliente . Todo esto de forma transparente para el cliente.

Localizadores Uniformes de Recursos (URLs)

Como vimos, las URLs estan compuestas por 3 partes, el protocolo (también llamado esquema), el nombre DNS de la máquina en donde se encuentra la página y un nombre local que indica de manera única la página específica (por lo general, sólo un nombre de archivo de la máquina en que reside) .

Este esquema de URL es abierto en el sentido de que es directo hacer que los navegadores utilicen múltiples protocolos para obtener diferentes tipos de recursos. De hecho, se han definido los URLs de varios otros protocolos comunes. Ademas del conocido http, podemos utilizar ftp, file para archivos locales, news para leer noticias, telnet, mailto para escribir mails, etc.

Sin estado y cookies

Cuando un cliente solicita una página Web, el servidor puede proporcionar información adicional junto con la página solicitada. Esta información puede incluir una cookie, que es un pequeño archivo (o cadena, de a lo mucho 4 KB). Los navegadores almacenan cookies ofrecidas en un directorio de cookies en el disco duro de la máquina del cliente, a menos que el usuario las haya deshabilitado. Las cookies son simplemente archivos o cadenas, no programas ejecutables. En principio, una cookie puede contener un virus, pero puesto que las cookies se tratan como datos, no hay una forma oficial de que los virus se ejecuten realmente y hagan daño.

Una cookie puede contener hasta cinco campo:

- Dominio: indica de dónde viene la cookie. Cada dominio puede almacenar hasta 20 cookies por cliente.
- Ruta es la estructura del directorio del servidor que identifica qué partes del árbol de archivos del servidor podrían utilizar la cookie. Por lo general es /, lo que significa el árbol completo.
- Contenido: toma la forma nombre = valor. Tanto nombre como valor pueden ser lo que el servidor desee. Este campo es donde se almacena el contenido de la cookie.
- Expira: especifica cuándo caduca la cookie. Si este campo está ausente, el navegador descarta la cookie cuando sale. Tal cookie se conoce como cookie **no persistente**. Si se proporciona una hora y una fecha, se dice que la cookie es **persistente** y se mantiene hasta que expira. Para eliminar una cookie del disco duro de un cliente, un servidor simplemente la envía nuevamente, pero con una fecha caducada.
- Seguro: puede establecerse para indicar que el navegador podría simplemente regresar la cookie a un servidor seguro. Esta característica se utiliza para comercio electrónico, actividades bancarias y otras aplicaciones seguras.

Justo antes de que un navegador solicite una página a un sitio Web, verifica su directorio de cookies para ver si el dominio al que está solicitando la página ya colocó alguna cookie. De ser así, todas las cookies colocadas por ese dominio se incluyen en el mensaje de solicitud. Cuando el servidor las obtiene, puede interpretarlas de la forma que desee.

Protocolo de Transferencia de Hipertexto

El protocolo de transferencia utilizado en World Wide Web es HTTP. Especifica cuáles mensajes pueden enviar los clientes a los servidores y qué respuestas obtienen. Cada interacción consiste en

una solicitud ASCII, seguida por una respuesta tipo MIME del RFC 822. Todos los clientes y servidores deben obedecer este protocolo. Se define en el RFC 2616.

Conexiones

La forma común en que un navegador contacta a un servidor es estableciendo una conexión TCP con el puerto 80 de la máquina del servidor, aunque este procedimiento no se requiere formalmente. El valor de utilizar TCP es que ni los navegadores ni los servidores tienen que preocuparse por los mensajes largos, perdidos o duplicados, ni por las confirmaciones de recepción.

A partir de HTML 1.1 las conexiones son **persistentes**, es decir, una vez establecida la conexión, el cliente puede realizar varias solicitudes, sin la necesidad de establecer una conexión nueva para cada una. También es posible enviar solicitudes en canalización, es decir, enviar la solicitud 2 antes de que la respuesta a la solicitud 1 haya llegado.

Métodos

HTTP se ha hecho intencionalmente más general de lo necesario con miras a las aplicaciones orientadas a objetos futuras. Por esta razón, se soportan otras operaciones, llamadas métodos, diferentes a las de solicitar una página Web. Cada solicitud consiste en una o más líneas de texto ASCII, y la primera palabra de la primera línea es el nombre del método solicitado. Para acceder a objetos generales, también están disponibles métodos adicionales específicos de objetos. Los nombres son sensibles a mayúsculas y minúsculas. Los métodos disponibles son:

- GET: solicita la lectura de una pagina Web
- HEAD: solicita la lectura del encabezado de una pagina. Se puede utilizar para obtener la fecha de la ultima modificación de la pagina, para indización por ejemplo.
- PUT: solicita el almacenamiento de una pagina. El cuerpo de la solicitud contiene la pagina, junto con datos de autenticación.
- POST: similar a PUT, solo que este método inserta los datos en algún sentido generalizado. Por ejemplo un mensaje a un grupo de noticias.
- DELETE: elimina la pagina. Se debe acompañar de datos de autenticación.
- TRACE: se utiliza en depuración. Indica al servidor que devuelva la ultima solicitud.
- CONNECT: no se utiliza. Se reserva para un uso futuro.
- OPTIONS: proporciona una forma para que el cliente consulte al servidor sobre sus propiedades o las de un archivo específico.

Cada solicitud obtiene una respuesta que consiste en una línea de estado, y posiblemente de información adicional (por ejemplo, toda o parte de una página Web). La línea de estado contiene un código de estado de tres dígitos que indica si la solicitud fue atendida, y si no, por qué. El primer dígito se utiliza para dividir las respuestas en cinco grupos mayores. En la práctica, los códigos significan:

- 1xx: no se utilizan con frecuencia.
- 2xx: significan que la solicitud se manejó de manera exitosa y que se regresa el contenido (si hay alguno).
- 3xx: indican al cliente que busque en otro lado, ya sea utilizando un URL diferente o en su propio caché (que se analiza posteriormente).
- 4xx: significan que la solicitud falló debido a un error del cliente, por ejemplo una solicitud

inválida o una página no existente.

- 5xx significan que el servidor tiene un problema, debido a un error en su código o a una sobrecarga temporal.

Encabezados de mensaje

A la línea de solicitud (por ejemplo, la línea con el método GET) le pueden seguir líneas adicionales que contienen más información. Éstas se llaman encabezados de solicitud. Esta información puede compararse con los parámetros de una llamada a procedimiento. Las respuestas también pueden tener encabezados de respuesta. Algunos encabezados pueden utilizarse en cualquier dirección. Los encabezados mas importantes son:

Encabezado	Tipo	Contenido
User-Agent	Solicitud	Información acerca del navegador y su plataforma
Accept	Solicitud	El tipo de páginas que el cliente puede manejar
Accept-Charset	Solicitud	Los conjuntos de caracteres que son aceptables para el cliente
Accept-Encoding	Solicitud	Las codificaciones de página que el cliente puede manejar
Accept-Language	Solicitud	Los idiomas naturales que el cliente puede manejar
Host	Solicitud	El nombre DNS del servidor
Authorization	Solicitud	Una lista de las credenciales del cliente
Cookie	Solicitud	Regresa al servidor una cookie establecida previamente
Date	Ambas	Fecha y hora en que se envió el mensaje
Upgrade	Ambas	El protocolo al que el emisor desea cambiar
Server	Respuesta	Información acerca del servidor
Content-Encoding	Respuesta	Cómo se codifica el contenido (por ejemplo, gzip)
Content-Language	Respuesta	El idioma natural utilizado en la página
Content-Length	Respuesta	La longitud de la página en bytes
Content-Type	Respuesta	El tipo MIME de la página
Last-Modified	Respuesta	La fecha y hora de la última vez que cambió la página
Location	Respuesta	Un comando para que el cliente envíe su solicitud a cualquier otro lugar
Accept-Ranges	Respuesta	El servidor aceptará solicitudes de rango de bytes
Set-Cookie	Respuesta	El servidor desea que el cliente guarde una cookie

FTP (File Transfer Protocol)

Es un protocolo de red para la transferencia de archivos entre sistemas conectados a una red TCP, basado en la arquitectura cliente-servidor. Desde un equipo cliente se puede conectar a un servidor para descargar archivos desde él o para enviarle archivos.

El servicio FTP es ofrecido utilizando normalmente el puerto de red 20 (para el envío de datos, es decir, archivos) y el 21 (para el envió de ordenes y control). Un problema básico de FTP es que está pensado para ofrecer la máxima velocidad en la conexión, pero no la máxima seguridad, ya que todo el intercambio de información, desde el login y password del usuario en el servidor hasta la transferencia de cualquier archivo, se realiza en texto plano sin ningún tipo de cifrado.

FTP permite copiar archivos de un sistema a otro, listar directorios, realizar tareas de gestión como eliminar archivos y cambiar nombres, etc.

Tiene un hermano menor llamado TFTP (Trivial FTP) que es un protocolo de transferencia muy simple. A menudo se utiliza para transferir pequeños archivos entre ordenadores en una red, como cuando un terminal X Window o cualquier otro cliente ligero arranca desde un servidor de red.

TELNET

Es un protocolo de red a otra máquina para manejarla remotamente. También es el nombre del programa informático que implementa el cliente. La máquina a la que se acceda debe tener un programa especial que reciba y gestione las conexiones. El puerto que se utilizado generalmente es el 23.

Sólo sirve para acceder en modo terminal, pero fue una herramienta muy útil para arreglar fallos a distancia, sin necesidad de estar físicamente en el mismo sitio que la máquina que los tenía.

También se usaba para consultar datos a distancia, como datos personales en máquinas accesibles por red, información bibliográfica, etc.

Su mayor problema es de seguridad, ya que todos los nombres de usuario y contraseñas necesarias para entrar en las máquinas viajan por la red como texto plano. Por esta razón dejó de usarse, casi totalmente, hace unos años, cuando apareció y se popularizó SSH, que puede describirse como una versión cifrada de telnet.

Firewalls

Los firewalls (servidores de seguridad) consisten en obligar a todo el trafico de entrada y salida de una red privada a que pase por un unico punto. En este punto se coloca un firewall que controla todo lo que ingresa y sale.

Este firewall puede consistir en un enrutador que contenga tablas con orígenes y destinos (Ips y puertos) aceptados, orígenes y destinos prohibidos y reglas sobre lo que debe hacer con los paquetes que entran y salen. De esta forma se pueden bloquear determinados puertos (TELNET, USENET, etc) y prohibir el intercambio de información con ciertas IP (de otros paises, etc.).

El firewall también puede tener una **puerta de enlace de aplicación.** Esta trabaja en capa de aplicación. Por ejemplo, puede examinar cada mensaje que sale o entra (analizando los campos de encabezado, el tamaño del mensaje o incluso en el contenido).

Aunque un firewall este perfectamente configurado, siempre tiene vulnerabilidades. Por ejemplo, un atacante puede falsificar la dirección de origen de los paquetes que enviá para atravesar el firewall. Un espiá puede encriptar documentos para enviarlos desde el interior de la LAN hacia afuera, de esta forma la puerta de enlace no puede ver lo que se envía. También existen los ataques por denegación de servicio (DoS) y los ataques por denegación de servicio distribuidos (DDoS).

NAT (Network Address Translation)

Es un mecanismo utilizado por encaminadores IP para intercambiar paquetes entre dos redes que se asignan mutuamente direcciones incompatibles (IP privadas y publicas por ejemplo). Consiste en convertir, en tiempo real, las direcciones utilizadas en los paquetes transportados. También es necesario editar los paquetes para permitir la operación de protocolos que incluyen información de direcciones dentro de la conversación del protocolo (las sumas de comprobación de la cabecera IP por ejemplo).

Su uso más común es permitir utilizar direcciones privadas para acceder a Internet. De esta manera simultáneamente sólo pueden salir a Internet con una IP tantos equipos como direcciones públicas se hayan contratado. Esto es necesario debido al progresivo agotamiento de las direcciones IPv4. Se espera que con el advenimiento de IPv6 no sea necesario continuar con esta práctica.

NAT puede trabajar de manera estatica (NAT 1:1), en el que una dirección IP privada se traduce a una dirección IP pública, y donde esa dirección pública es siempre la misma. Esto le permite a un host, como un servidor Web, el tener una dirección IP de red privada pero aún así ser visible en Internet. Para ello usa la técnica llamada port forwarding.

NAT también puede trabajar de manera dinámica, en la que una dirección IP privada se mapea a alguna dirección de un pool de IP publicas. Mientras exista este mapeo la IP publica se marca como en uso y se mantiene una entrada en una tabla para realizar las traducciones. Cuando el host correspondiente a la IP privada no la necesita mas, esta entrada se elimina y la IP publica vuelve al pool para ser utilizada por otro host de la red privada. Esto permite aumentar la seguridad de una red dado que enmascara la configuración interna de una red privada, lo que dificulta a los hosts

externos de la red el poder ingresar a ésta. Para este método se requiere que todos los hosts de la red privada que deseen conectarse a la red pública posean al menos una IP pública asociadas.

PAT (Port Address Translation)

Es una característica del estándar NAT, que traduce conexiones TCP y UDP hechas por un host y un puerto en una red privada a otra dirección y puerto de la red publica (Internet). Permite que una sola dirección IP publica sea utilizada por varias máquinas de la intranet. Con PAT, una IP publica puede responder hasta a ~64000 direcciones privada. PAT Se apoya en el hecho de que el puerto de origen carece de importancia para la mayoría de los protocolos. Igual que NAT, se sitúa en la frontera entre la red interna y externa, y realiza cambios en la dirección del origen y del receptor en los paquetes de datos que pasan a través de ella. Los puertos (no las IP), se usan para designar diferentes hosts en la red privada.

Cuando un host del intranet manda un paquete hacia fuera, el servicio NAT remplaza la IP interna con la nueva IP del propio servicio tomada de un pool de direcciones publicas. Luego asigna a la conexión un puerto de la lista de puertos disponibles, inserta el puerto en el campo apropiado del paquete de datos y envía el paquete. El servicio NAT crea una entrada en su tabla de direcciones IP internas, puertos internos, IP externas y puertos externos. A partir de entonces, todos los paquetes que provengan del mismo hosts serán traducidos con los mismos puertos y dirección. El receptor del paquete utilizará los IP y puerto recibidos para responder. El traductor revisara la tabla de traducciones, colocara la nueva dirección y el nuevo puerto de destino en el paquete y éste será enviado por la red interna.

Proxy

Es un programa o dispositivo que realiza una acción en representación de otro, esto es, si una hipotética máquina A solicita un recurso a una C, lo hará mediante una petición a B; C entonces no sabrá que la petición procedió originalmente de A.

Su finalidad más habitual es la de servidor proxy, que consiste en interceptar las conexiones de red que un cliente hace a un servidor de destino. De ellos, el más famoso es el servidor proxy web (Comúnmente conocido solo como proxy), pero también existen proxies para otros protocolos, como el proxy de FTP.

Hay dos tipos de proxys atendiendo a quien es el que quiere implementar la política del proxy:

- proxy local: En este caso el que quiere implementar la política es el mismo que hace la petición. Suelen estar en la misma máquina que el cliente que hace las peticiones. Son muy usados para que el cliente pueda controlar el tráfico y pueda establecer reglas de filtrado que por ejemplo pueden asegurar que no se revela información privada.
- proxy externo: El que quiere implementar la política del proxy es una entidad externa. Se suelen usar para implementar cacheos, bloquear contenidos, control del tráfico, etc.

Entre las ventajas de utilizar un proxy podemos nombrar:

- Control: todo lo que se realiza pasa por el proxy.
- Ahorro: solo el proxy esta preparado para realizar ciertos trabajos.
- Velocidad: por ejemplo utilizando cache en el proxy.
- Filtrado: e proxy decide que dejar pasar y que no.
- Modificación: un proxy puede falsificar información, o modificarla siguiendo un algoritmo.

- Anonimato: muchos usuarios pueden utilizar la identidad del proxy.

Tambien presenta desventajas, como la intromisión, la incoherencia que puede ocasionar el uso de cache y las irregularidades que generan esta “falsificación de la identidad (en TCP/IP trae problemas).