Respuestas de Comunicación de Datos II

Indice

Contenido

Introducción al nivel de red

Mencione y explique brevemente las funciones principales del nivel de red

Provee transporte de paquetes de un equipo de usuario a otro, independizándolos de la tecnología y topología de la red que soporta la comunicación.

- Identificación de los equipos que se conectan a la red (ADDRESSING)
- Secuencia de nodos a recorrer por un paquete para llegar a destino con un costo mínimo (RUTEO)
- Establecimiento, terminación y control de conexiones a nivel de red
- Mecanismos para evitar que se congestione parte o la totalidad de la red
- Control de la tasa de envío de un emisor
- Tarifas a cobrar a los equipos por el uso de la red
- Seguridad (autenticación, cifrado)
- Calidad de servicio (QoS)

Describa y compare las posibles organizaciones internas del nivel de red: circuito virtual y datagram

1.

Item	Datagram	Circuito Virtual
Establecimiento de circuito	No	Sí
Recursos en los routers	Sólo ruteo	Reserva de recursos para cada CV
Ruteo	Por paquete	Por CV
Efecto de fallas en la red	Pérdida temporaria de paquetes en la zona afectada	Caída de los CV en la zona afectada
Control de congestión	Complicado	Relativamente simple
Complejidad	En los hosts (control de errores, etc)	En la red
Overhead	Overhead en cada paquete	Overhead inicial (setup)
Confianza	El usuario confía en sí mismo	El usuario confía en la red

Qué es y en qué consiste MPLS (Multi Protocol Label Switching). Aspectos que permite mejorar respecto de una red orientada a datagram.

MPLS es un protocolo de transmisión de datos orientado a la conexión. Se enfocó en agregar una etiqueta en frente de cada paquete y realizar el enrutamiento con base en ella y no con base en la dirección de destino. Hacer que la etiqueta sea un índice de una tabla provoca que encontrar la línea correcta de salida sea una simple cuestión de buscar en una tabla. Al utilizar esta técnica, el enrutamiento puede llevarse a cabo de manera muy rápida y cualesquier recursos necesarios pueden reservarse a lo largo de la ruta.

Cosas básicas de un servicio orientado a la conexión.

Describa los componentes y conceptos más importantes de MPLS (Multi Protocol Label Switching), LER, LSR, LSP, etc. Explique la operación de MPLS.

FEC (Forward Equivalence Class): conjunto de paquetes a ser enviados por la misma ruta, puede depender del ip origen y del tipo de servicio especificado

Label: identificador local de FEC

Protocolos de distribución de labels:

- LDP (Label Distribution Protocol)
- BGP (Border gateway Protocol –modificado-)
- RSVP (Resource Reservation Protocol –modificado-)

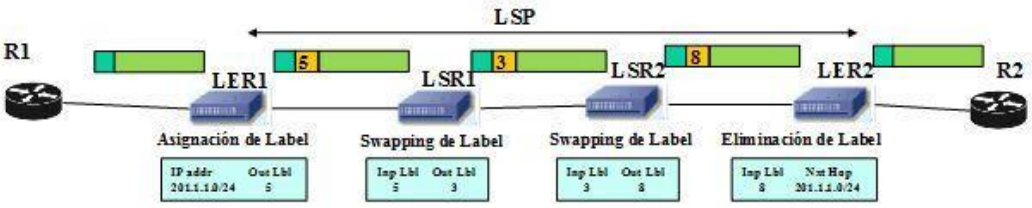
LSP (Label-Switched Path): camino determinado por los labels

LER (Label Edge Router): Router encargado de asignar el label (ruteo), son los routers de frontera con otras redes.

LSR (Label Switching Router): router o switch encargado del switching, son internos a la red MPLS

Operación de MPLS

1-LER1 recibe un paquete de R1 para R2
2-No tiene todavía un label. Busca el next hop (LSR1) de acuerdo a la dirección Ip de destino
3-Envía un label request a LSR1
4-Cada nodo hasta el destino (LER2) recibe el label request que se va propagando (LDP)
5-LER2 genera un label para la FEC, y lo propaga hacia LSR2 (label distribution) (LDP)
6-Cada nodo completa su tabla
7-LER1 agrega label y envía según su tabla
8-LSR1 realiza swapping de labels y envía
. . .
9-LER2 elimina el label y envía según IP



Diferencias entre una red MPLS (Multi Protocol Label Switching) respecto de una red orientada a circuito virtual y una orientada a datagram

Una diferencia con respecto a los circuitos virtuales tradicionales es el nivel de agregación. Ciertamente es posible que cada flujo tenga su propio conjunto de etiquetas a través de la subred. Sin embargo, es más común que los enrutadores agrupen múltiples flujos que terminan en un en-rutador o una LAN particulares y utilizan una sola etiqueta de ellos. Se dice que los flujos que están agrupados en una sola etiqueta pertenecen a la misma FEC (clase de equivalencia de reenvío).

Esta clase cubre no sólo a dónde van los paquetes, sino también su clase de servicio (en el sentido de los servicios diferenciados), debido a que todos sus paquetes se tratan de la misma forma para propósitos de reenvío.

Con el enrutamiento de circuitos virtuales tradicional no es posible agrupar en el mismo identificador de circuitos virtuales varias rutas diferentes con diferentes puntos finales, debido a que podría no haber forma de distinguirlas en el destino final. Con MPLS, los paquetes aún contienen su dirección de destino final, además de la etiqueta, a fin de que al final de la red de MPLS pueda eliminarse la etiqueta y que el reenvío pueda continuar de la forma normal, utilizando la dirección de destino de la capa de red.

Una diferencia principal entre MPLS y los diseños de circuitos virtuales convencionales es la forma en que está construida la tabla de reenvío. En las redes de circuitos virtuales tradicionales, cuando un usuario desea establecer una conexión, se inicia un paquete de configuración en la red para crear la ruta y crear las entradas

de la tabla de reenvío. MPLS no funciona de esa forma porque no hay fase de configuración para cada conexión (pues eso podría romper con la operación de mucho software existente en Internet).

- Un CV equivale a un flujo desde un origen hasta un destino
  Una FEC puede involucrar múltiples flujos entre diferentes pares origen-destino
- En CV, el único medio de reenvío de paquetes es el switching
  En MPLS, se reenvía además por routing IP
- CV soporta un único nivel de labels
  MPLS soporta múltiples niveles de labels
- Un dominio MPLS (de LER a LER) no necesariamente va de origen a destino
  En MPLS, los usuarios no se ven involucrados en ningún setup, sólo envían paquetes.

Explique y de ejemplos de las técnicas de "overlay" y "translación" utilizadas para la interconexión de redes. Ventajas y desventajas de cada una.

Una red superpuesta es una red virtual de nodos enlazados lógicamente, que está construida sobre una o más redes subyacentes (*underlying network* en inglés). Su objetivo es implementar servicios de red que no están disponibles en la red/es subyacente/s. Las redes superpuestas pueden apilarse de forma que tengamos capas que proporcionen servicios a la capa superior.

Ventajas

- No se necesita agregar nuevo equipamiento o modificar el software o protocolos existentes. Se debe agregar nuevo software.
- No tiene que deployarse en cada nodo
  - No todos los nodos necesitan/quieren servicio overlay todo el tiempo
  - Las redes overlay pueden ser muy pesadas para algunos nodos (memoria, ancho de banda)
  - Las redes overlay pueden tener propiedades de seguridad poco claras
  - Las redes overlay pueden no escalarse...

Desventajas

- Agrega overhead (una capa nueva, encabezado, a veces redundante)
- Agrega complejidad (al tener más capas de funcionalidad es más posible tener interacciones no deseadas entre capas)

En la traslación los routers conecten más de un protocolo.

Mencione 5 diferencias entre las distintas redes físicas y explique cómo las resolvería un protocolo de interred como IP

Aspecto	Algunas posibilidades
Servicio ofrecido	Sin conexiones, orientado a conexiones
Protocolos	IP, IPX, SNA, ATM, MPLS, AppleTalk, etc.
Direccionamiento	Plano (802) o jerárquico (IP)
Multidifusión	Presente o ausente (también difusión)
Tamaño de paquete	Cada red tiene su propio máximo
Calidad del servicio	Puede estar presente o ausente; muchos tipos diferentes
Manejo de errores	Entrega confiable, ordenada y desordenada
Control de flujo	Ventana corrediza, control de tasa, otros o ninguno
Control de congestión	Cubeta con goteo, paquetes reguladores, etc.
Seguridad	Reglas de confidencialidad, encriptación, etc.
Parámetros	Diferentes terminaciones de temporizador, especificaciones de flujo, etc.
Contabilidad	Por tiempo de conexión, por paquete, por byte, o sin ella

Figura 5-43. Algunas de las muchas maneras en que pueden diferir las redes.

Por ejemplo cuando se quiere conectar una red orientada a la conexión y otra que no, se requerirá de un reordenamiento.
También se necesitarán conversiones de direcciones, lo que podría requerir algún tipo de sistema de directorio.
El paso de paquetes multidifusión a través de una red que no reconoce la multidifusión requiere la generación de paquetes individuales para cada destino.

Explique detalladamente y con ejemplos la técnica de descubrimiento de MTU

Por razones de eficacia, generalmente es preferible que la información intercambiada entre dispositivos esté contenida en datagramas de tamaño máximo. Este tamaño depende de la ruta seguida por los datagramas y es igual al tamaño más grande autorizado por el conjunto de enlaces recorridos. Se le llama PMTU, Path Maximum Transmission Unit (Unidad de transferencia de tamaño máximo sobre la trayectoria).

Inicialmente, el dispositivo supone que el PMTU de una cierta trayectoria es igual al MTU del enlace al que está directamente conectado. Si sucede que los paquetes enviados por ese camino exceden el tamaño máximo autorizado por un enlace intermedio, el enrutador correspondiente eliminará esos paquetes y enviará un mensaje de error ICMPv6 tipo “paquete demasiado grande” donde indicará el MTU aceptable. Con esa información, el dispositivo reducirá el PMTU apropiado para esa ruta.

Es posible que se deban efectuar varias iteraciones antes de obtener el PMTU que permita que los paquetes lleguen a su destino sin exceder el MTU de cada enlace recorrido. El protocolo IPv6 garantiza que el MTU de todos los enlaces no puede ser menor de 1,280 octetos, por lo que éste es el umbral mínimo para el PMTU.

Dado que este protocolo puede sufrir pérdidas de paquetes, se deja a las capas superiores gestionar la fiabilidad de la comunicación y retransmitir paquetes si es necesario.

Si bien la determinación del PMTU se hace esencialmente durante los primeros intercambios entre los dispositivos comunicantes, es posible que pueda reactivarse durante una comunicación activa si, debido a un cambio de ruta, se debe recorrer un enlace con mayores restricciones de tamaño.

El emisor verifica también que el PMTU no ha aumentado enviando esporádicamente un paquete más grande. Si éste cruza la red sin problemas, el valor del PMTU se incrementará.

Cabe señalar que el algoritmo de descubrimiento de PMTU funciona igual para las comunicaciones punto a punto, o multipunto. En este último caso, el PMTU será el mínimo permitido para el conjunto de caminos hacia cada sitio destinatario del grupo de difusión.

Explique detalladamente y con ejemplos la técnica de fragmentación

Los diseñadores de redes no están en libertad de escoger cualquier tamaño máximo de paquetes que deseen. Surge un problema obvio cuando un paquete grande quiere viajar a través de una red cuyo tamaño máximo de paquete es demasiado pequeño. Una solución es asegurar que no ocurra el problema.

En otras palabras, la interred debe usar un algoritmo de enrutamiento que evite el envío de

Paquetes a través de redes que no pueden manejarlos.

Básicamente, la única solución al problema es permitir que las puertas de enlace dividan los paquetes en fragmentos, enviando cada paquete como paquete de interred individual. Las redes de conmutación de paquetes también tienen problemas al unir nuevamente los fragmentos.

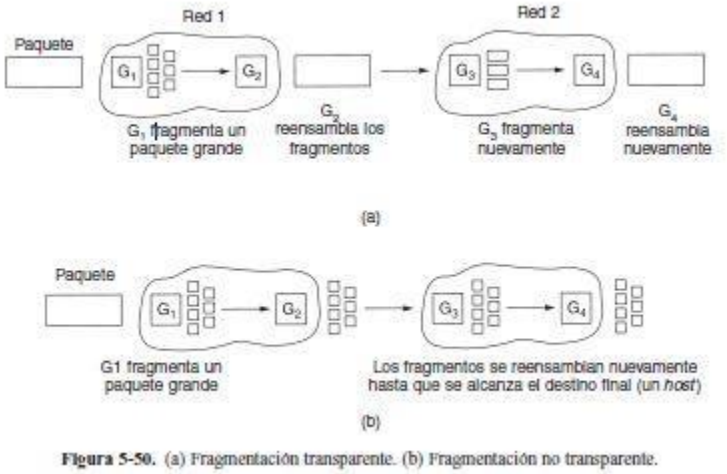


Figura 5-50. (a) Fragmentación transparente. (b) Fragmentación no transparente.

Existen dos estrategias opuestas para recombinar los fragmentos y recuperar el paquete original.

La primera es hacer transparente la fragmentación causada por una red de “paquete pequeño” a las demás redes subsiguientes por las que debe pasar el paquete para llegar a su destino final.

Esta opción se muestra en la figura 5-50(a). Con este método, la red de paquete pequeño tiene puertas de enlace (lo más probable es que sean enrutadores especializados) que interactúan con otras redes. Cuando un paquete de tamaño excesivo llega a una puerta de enlace, ésta lo divide en fragmentos. Todos los fragmentos se dirigen a la misma puerta de enlace de salida, donde se recombinan las piezas. De esta manera se ha hecho transparente el paso a través de la red de paquete pequeño. Las redes subsiguientes ni siquiera se enteran de que ha ocurrido una fragmentación.

Las redes ATM, por ejemplo, tienen hardware especial para proporcionar fragmentación

Transparente de paquetes en celdas y luego re ensamblar las celdas en paquetes. En el mundo ATM, a la fragmentación se le llama segmentación; el concepto es el mismo, pero algunos de los detalles son diferentes.

La fragmentación transparente es sencilla, pero tiene algunos problemas. Por una parte, la puerta de enlace de salida debe saber cuándo ha recibido todas las piezas, por lo que debe incluirse un campo de conteo o un bit de “fin de paquete” en cada paquete. Por otra parte, todos los paquetes deben salir por la misma puerta de enlace. Al no permitir que algunos fragmentos sigan una ruta al destino final, y otros fragmentos una ruta distinta, puede bajar un poco el desempeño. Un último problema es la sobrecarga requerida para re ensamblar y volver a fragmentar repetidamente un paquete grande que pasa a través de una serie de redes de paquete pequeño. ATM requiere fragmentación transparente.

La otra estrategia de fragmentación es abstenerse de recombinar los fragmentos en las puertas de enlace intermedias. Una vez que se ha fragmentado un paquete, cada fragmento se trata como si fuera un paquete original. Todos los fragmentos pasan a través de la puerta de enlace (o puertas de enlace) de salida, como se muestra en la figura 5-50(b). La recombinación ocurre sólo en el host de destino. IP funciona de esta manera.

La fragmentación no transparente también tiene algunos problemas. Por ejemplo, requiere que “todos” los hosts sean capaces de hacer el re ensamble. Otro problema es que, al fragmentarse un paquete grande, aumenta la sobrecarga total, pues cada fragmento debe tener un encabezado. En tanto que en el primer método la sobrecarga desaparece en cuanto se sale de la red de paquete pequeño, en este método la sobrecarga permanece durante el resto de la travesía. Sin embargo, una ventaja de este método es que ahora pueden usarse varias puertas de enlace de salida, lográndose un mejor desempeño. Por supuesto, si se está usando el modelo de circuito virtual concatenado, esta ventaja no es de ninguna utilidad.

Cuando se divide un paquete, los fragmentos deben numerarse de tal manera que el flujo de datos original pueda reconstruirse.

Compare las técnicas de fragmentación y de descubrimiento de MTU. Ventajas y desventajas de cada una de ellas respecto de la otra.

Descubrimiento MTU	**Fragmentación**
Homogeniza el tamaño de los paquetes	Los paquetes son fragmentados o unidos(según la técnica sea transparente o no), según el mtu de la

	red
Se limita al menor tamaño en el conjunto de redes por las que pasa (incluso si es muy pequeño respecto a los demás)	El tamaño del paquete depende de la red en donde se encuentre
Demora en el descubrimiento del MTU, necesita un tiempo de inicialización	Demora por la fragmentación.
Overhead (mayor!)	Inserción de overhead, se añadirán encabezados a cada paquete conforme estos se vayan dividiendo

Describa el direccionamiento IPv4 classful (estructura, tipos de direcciones, direcciones especiales, privadas, etc.)

Los desarrolladores del protocolo IP reconocieron que las organizaciones tenían diferentes tamaños y por lo tanto sería necesario un número variable de direcciones IP en Internet. Diseñaron un sistema en el que el espacio de direcciones IP sería dividido en clases, cada una de las cuales contenía una porción de la dirección total y se dedicaban a usos específicos. Algunas se dedicarían a las grandes redes en Internet, mientras que otras serían para organizaciones más pequeñas, y otras reservadas para propósitos especiales.

Hay cinco clases en el sistema "classful", que corresponden a las letras de la A a la E.

Clase	Fracción del espacio total de direcciones.	Numero de bits del Id de Red	Numero de bits del Id de host	Uso
A	½	8	24	Direccionamiento unicast para organizaciones muy grandes con cientos de miles de millones de hosts que conectar a Internet.
B	¼	16	16	Direccionamiento unicast para organizaciones de medias a grandes con varios cientos de miles de hosts que conectar a Internet
C	1/8	24	8	Direccionamiento unicast para organizaciones mas pequeñas con no mas de 250 hosts que conectar a internet
D	1/16	n/a	n/a	Multidifusión IP
E	1/16	n/a	n/a	Reservado para uso experimental

Es justo recordar también las numerosas ventajas del sistema "classful" desarrollado hace más de 25 años:

Sencillez y claridad: Hay sólo unas pocas clases para elegir y es muy fácil de entender cómo se dividen las direcciones. La distinción entre las clases es clara y evidente. Las divisiones entre los ID de red y el ID de hosts en las clases A, B y C están en los límites de octeto, y esto hace fácil decir cuál es el identificador de red de cualquier dirección.

Razonable flexibilidad: Los tres niveles de "granularidad" coinciden razonablemente bien con los tamaños de las organizaciones grandes, medianas y pequeñas. El sistema original preveía capacidad suficiente para manejar la tasa de crecimiento esperado de la Internet en el momento.

Facilidad de enrutamiento: La clase de la dirección se codifica justo en la dirección para que le sea fácil a los routers saber qué parte de cualquier dirección es el identificador de red y qué parte es la ID de host. No hay necesidad de información anexa, tal como la máscara de subred.

Direcciones reservadas: Algunas direcciones están reservadas para propósitos especiales. Esto incluye no sólo las clases D y E, sino también rangos especiales de direcciones reservadas para el direccionamiento "privado".

El esquema de direccionamiento IP "classful" divide el espacio de direcciones IP en cinco clases de diferentes tamaños, de la A a la E. Las clases A, B y C son las más importantes, designadas para las direcciones unicast convencionales y comprenden 7/8vos del espacio total de direcciones. Clase D está reservado para la multidifusión IP, y la clase E para uso experimental.

Problemas con el esquema:

- Codificar la red en la dirección IP implica que si un host cambia de red, cambiará su dirección (IP Mobility).
- Prefijos de longitud fija, provoca un uso ineficiente en el espacio de direcciones.
- Crecimiento acelerado de la Internet, evidencia la falta de escalabilidad del esquema de direccionamiento (Agotamiento de clases B, incremento de tamaño de tablas de ruteo al utilizar direcciones de clase C)

Direcciones no asignables

- La dirección 255.255.255.255 se utiliza para indicar broadcast en la propia red.
- La dirección 0.0.0.0 identifica al host actual.
- Las direcciones con el campo host todo a ceros identifican redes.
- La dirección con el campo host todo a unos se utiliza como dirección broadcast dentro de la red.
- La dirección con el campo red todo a ceros identifica a un host en la propia red.
- La dirección 127.0.0.1 se utiliza para pruebas loopback.
- Las redes 127.0.0.0, 128.0.0.0, 191.255.0.0, 192.0.0.0 y el rango de 240.0.0.0 en adelante (clase E) están reservados y no deben utilizarse.
- Las redes 10.0.0.0 (clase A), 172.16.0.0 a 172.31.0.0 (clase B) y 192.168.0.0 a 192.168.255.0 (clase C) están reservadas para redes privadas ('intranets')

Describa brevemente la estructura y generalidades de las direcciones IPv6

El IPv6 mantiene las buenas características del IP, descarta y reduce las malas, y agrega nuevas donde se necesitan. En general, IPv6 no es compatible con IPv4, pero es compatible con todos los demás protocolos

Internet, incluidos TCP, UDP, ICMP, IGMP, OSPF, BGP y DNS, a veces con algunas pequeñas modificaciones principalmente para manejar direcciones más grandes).

Por principio, y lo más importante, el IPv6 tiene direcciones más grandes que el IPv4; son de 16 bytes de longitud, lo que resuelve el problema que se buscaba resolver: proporcionar una cantidad prácticamente ilimitada de direcciones Internet.

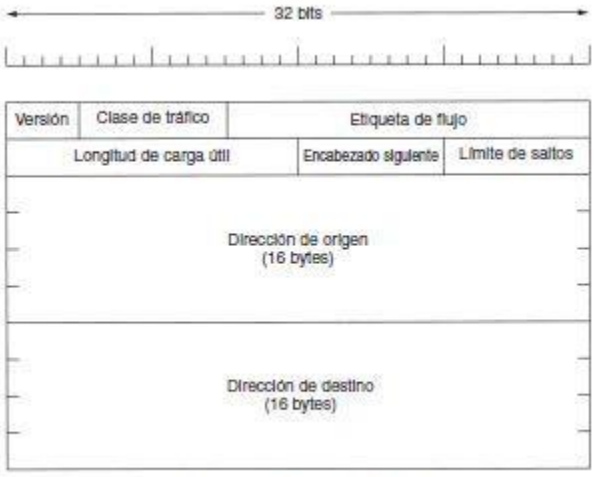


Figura 5-68. Encabezado fijo del IPv6 (obligatorio).

La segunda mejora principal del IPv6 es la simplificación del encabezado, que contiene sólo 7 campos (contra 13 en el IPv4). Este cambio permite a los enrutadores procesar con mayor rapidez los paquetes y mejorar, por tanto, la velocidad real de transporte.

La tercera mejora importante fue el mejor apoyo de las opciones. Este cambio fue esencial con el nuevo encabezado, pues campos que antes eran obligatorios ahora son opcionales. Además, es diferente la manera de representar las opciones, haciendo más sencillo que los enrutadores hagan caso omiso de opciones no dirigidas a ellos. Esta característica mejora el tiempo de procesamiento de los paquetes.

Con todo, algunos de los campos faltantes del IPv4 en ocasiones son necesarios, por lo que el IPv6 introdujo el concepto de encabezado de extensión (opcional). Estos encabezados pueden usarse para proporcionar información extra, pero codificada de una manera eficiente. Hay seis tipos de encabezados de extensión definidos actualmente, que se listan en la figura 5-69. Todos son opcionales, pero si hay más de uno, deben aparecer justo después del encabezado fijo, y de preferencia en el orden listado.

Encabezado de extensión	Descripción
Opciones salto por salto	Información diversa para los enrutadores
Opciones de destino	Información adicional para el destino
Enrutamiento	Ruta total o parcial a seguir
Fragmentación	Manejo de fragmentos de datagramas
Autenticación	Verificación de la identidad del emisor
Carga útil de seguridad encriptada	Información sobre el contenido encriptado

Una cuarta área en la que el IPv6 representa un avance importante es la seguridad.

Por último, se ha puesto mayor atención en la calidad del servicio.

Explique con ejemplos el método utilizado para representar direcciones IPV6

Representación hexadecimal

Los 128 bits se dividen en grupos de 16

Cada grupo se expresa como 4 dígitos hexadecimales

Los grupos se separan con ":"

21BA:00D2:0000:3F4D:0C34:00FF:FE28:9C5A

Supresión de ceros

Se suprimen ceros al comienzo de cada bloque

Cada bloque debe tener al menos un digito hexadecimal

21BA:D2:0:3F4D:C34:FF:FE28:9C5A

Compresión de ceros

Grupos de ceros contiguos se reemplazan por "::"

Solo es posible comprimir grupos completos y no partes

Solo se puede utilizar una única vez en una dirección

FE80:0:0:0:2AA:FF:FE9A:9C5A valido FE80::2AA:FF:FE9A:9C5A

FF02:0:0:0:0:0:0:2 valido FF02::2

FE80:0:0:2AA:0:0:0:9C5A no valido FE80::2AA::9C5A

FE80:60:0:0:0:0:0:9C5A no valido FE80:6::9C5A

En situaciones donde se usan direcciones IPv4 e IPv6:

::192.168.1.2 ó 0:0:0:0:0:0:192.168.1.2

Que tipos de direcciones contempla IPv6? Explique para qué se utiliza cada uno de ellos.

Una dirección IPv6 identifica un punto de acceso (interfaz) a la red.

Una interfaz puede tener asignadas una o más direcciones, de cualquier tipo.

Una dirección unicast puede estar asignada a dos o más interfaces.

Un equipo (host, router) es identificado por una dirección unicast asignada a cualquiera de sus interfaces.

Los valores todos ceros o todos unos son válidos para el campo red o host.

La arquitectura de las direcciones IPv6 incluye el scope (alcance) de la dirección.

A un link, se le asigna una subred (dirección de red con un prefijo).

Pueden asignarse múltiples subredes a un link (multinetting).

Tipos de direcciones

Unicast

Identifican una interfaz de manera unívoca dentro del scope de la dirección

Multicast

Identifican múltiples interfaces de conexión a la red. Un datagram dirigido a una dirección multicast se entrega a todos los miembros.

Anycast

Identifican múltiples interfaces de conexión a la red. Un datagram dirigido a una dirección multicast se entrega sólo a uno de los miembros

NO se definen direcciones broadcast

Asignación	Pref. binario	Pref. hexa	Fracción
Unassigned	0000 0000	::0/8	1/256
Reserved	0000 001	0200::/7	1/128
Global unicast	001	2000::/3	1/8
Link-local unicast	1111 1110 10	FE80::/10	1/1024
Reserved (formely Site-local unicast)	1111 1110 11	FEC0::/10* * deprecated	1/1024
Local IPv6 address	1111 110	FC00::/7	
Private administration	1111 1101	FD00::/8	
Multicast	1111 1111	FF00::/8	1/256

Tipos de direcciones unicast

Direcciones globales sumarizables (Global Unicast)

Prefijo: 001 hex: 2000::/3

Direcciones locales nivel link (Link Local Unicast)

Prefijo: 1111 1110 10 hex:FE80::/10

Direcciones locales a nivel site (Site Local Unicast)

Se discontinua su uso (Deprecating Site Local Addresses, RFC 3879, sept. 2004)

Causas

Ambigüedad de direcciones en hosts multihomed o routers

Diferentes definiciones de sites reemplazadas por

Direcciones especiales (prefijo 000)

Dirección no especifica (0:0:0:0:0:0:0:0)

Dirección de loopback (0:0:0:0:0:0:0:1)

Direcciones para compatibilidad (prefijo 000)

IPv4 compatibles (0:0:0:0:0:0:w.x.y.z) (ya no utilizadas)

IPv4 mapeadas (0:0:0:0:0:FFFF:w.x.y.z)

Ruteo

Explique en qué consiste el ruteo por circuito virtual y por datagram. Ventajas y desventajas. Ejemplo concreto de su uso.

Ruteo por datagram: no orientado a la conexión, los paquetes se mandan independientes uno de otro. Cada paquete contiene en un encabezado agregado por la capa de red, la dirección completa de origen y destino para que cualquier router lo pueda encaminar correctamente.

Ruteo por circuito virtual: orientado a la conexión, se establece una conexión (circuito virtual) entre el origen y el destino, cada paquete con los mismos datos de origen y circuito sigue el mismo camino. Cada router debería tener la habilidad de intercambiar etiquetas de rutas, para lograr que los routers intermedio tengan información extra, ya que no podrán acceder directamente a las direcciones de origen para decidir porque circuito enviar al paquete.

Asunto	Subred de datagramas	Subred de circuitos virtuales
Configuración del circuito	No necesaria	Requerida
Direccionamiento	Cada paquete contiene la dirección de origen y de destino	Cada paquete contiene un número 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 manera independiente	Ruta escogida cuando se establece el CV; todos los paquetes siguen esta ruta
Efecto de fallas del enrutador	Ninguno, excepto para paquetes perdidos durante una caída	Terminan todos los CVs que pasan a través del enrutador
Calidad del servicio	Difícil	Fácil si se pueden asignar suficientes recursos por adelantado para cada CV
Control de congestión	Difícil	Fácil si pueden asignarse por adelantado suficientes recursos a cada CV

Internet ofrece servicio de capa de red no orientado a la conexión; las redes ATM ofrecen servicio de capa de red orientado a la conexión. Sin embargo, es interesante hacer notar que conforme las garantías de calidad del servicio se están volviendo más y más importantes, Internet está evolucionando.

Explique en qué consiste el ruteo “hop by hop” y “source routing”. Ventajas y desventajas. Ejemplo concreto de su uso.

Hop-by-hop es un método común de encaminamiento en redes en las que hay nodos intermedios entre la fuente y el destino, va a la dirección del siguiente nodo principal hasta el punto de destino en la lista. Así que cuando un paquete de datos llega a un nodo en particular, usa la tabla de rutas para encontrar la dirección del siguiente nodo. Una vez que llega a ese nodo, de nuevo usa la tabla de encaminamiento para la dirección del siguiente salto, y así sucesivamente, hasta llegar al destino final.

Una ventaja de este algoritmo es que es muy adaptativo. Como el camino del paquete se va decidiendo paso a paso, si llegado el momento uno de los links se cae, se podrá seguir por algún camino alternativo, siempre y cuando este exista.

Otra ventaja es la transparencia. Cada router solo se encargara de decidir el paso siguiente de los paquetes, delegando el resto del trabajo al resto de los routers

Source routing es un método de ruteo en el cual la información para realizar el ruteo es proveída por la fuente del mensaje a transmitir. Esta especificará la ruta exacta(o parte de ella) que deberá seguir el paquete a través de la red hasta su destino. Entre otras cosas esto permite que se pueda hacer control de congestión sobre la red, forzando a que los paquetes sigan las rutas más "libres". Otra ventaja es que puede utilizarse como un método para debuggear la red.
Como desventaja, si llegara a caerse uno de los links necesarios para enviar el paquete, este no llegaría a destino, porque no se especifica una ruta alternativa.

Ejemplo: LSRR, es una opción de encabezado de IP

Cómo se clasifican los algoritmos de ruteo según qué nodo o nodos toman las decisiones de ruteo? Dé un ejemplo de cada clase.

Pueden hacer más tolerantes a cambios en la subred tales como variaciones en el tráfico, incremento del retardo o fallas en la topología. El encaminamiento dinámico o adaptativo se puede clasificar a su vez en tres categorías, dependiendo de dónde se tomen las decisiones y del origen de la información intercambiada:

Adaptativo centralizado. Todos los nodos de la red son iguales excepto un nodo central que es quien recoge la información de control y los datos de los demás nodos para calcular con ellos la tabla de encaminamiento. Este método tiene el inconveniente de que consume abundantes recursos de la propia red.

Adaptativo distribuido. Este tipo de encaminamiento se caracteriza porque el algoritmo correspondiente se ejecuta por igual en todos los nodos de la subred. Cada nodo calcula continuamente la tabla de encaminamiento a partir de dicha información y de la que contiene en su propia base de datos. A este tipo pertenecen dos de los más utilizados en Internet que son los algoritmos por vector de distancias y los de estado de enlace.

Adaptativo aislado. Se caracterizan por la sencillez del método que utilizan para adaptarse al estado cambiante de la red. Su respuesta a los cambios de tráfico o de topología se obtiene a partir de la información propia y local de cada nodo. Un caso típico es el encaminamiento "por inundación" cuyo mecanismo consiste en reenviar cada paquete recibido con destino a otros nodos, por todos los enlaces excepto por el que llegó.

Cómo se clasifican los algoritmos de ruteo según la estrategia (cómo se adaptan a los cambios) de ruteo? Dé un ejemplo de cada clase

No adaptativos: No tienen en cuenta el estado de la subred al tomar las decisiones de encaminamiento. Las tablas de encaminamiento de los nodos se configuran de forma manual y permanecen

inalterables hasta que no se vuelve a actuar sobre ellas. Por tanto, la adaptación en tiempo real a los cambios de las condiciones de la red es nula.

El cálculo de la ruta óptima es también off-line por lo que no importa ni la complejidad del algoritmo ni el tiempo requerido para su convergencia. Ej.: algoritmo de Dijkstra.

Estos algoritmos son rígidos, rápidos y de diseño simple, sin embargo son los que peores decisiones toman en general, difícil de mantener ante cambios.

Dinámicos: Pueden ser más tolerantes a cambios en la subred tales como variaciones en el tráfico, incremento del retardo o fallas en la topología

Este tipo de algoritmos no pueden ser demasiado complejos ya que son implementados en los routers y deben ejecutarse en tiempo real con recursos de CPU y la memoria con que el router dispone.

Cómo se clasifican los algoritmos de ruteo según el origen de la información que utilizan los nodos para tomar las decisiones de ruteo?. Dé un ejemplo de cada clase

Local: flooding.

Nodos adyacentes: distancie vector.

Todos los nodos: link state.

Mencione y explique brevemente cada uno de los componentes de la función de ruteo y cómo se relacionan.

Protocolo de ruteo: Un protocolo de enrutamiento es un conjunto de procesos, algoritmos y mensajes que se usan para intercambiar información de enrutamiento y completar la tabla de enrutamiento con la selección de las mejores rutas del protocolo

De enrutamiento.

El propósito de un protocolo de enrutamiento incluye:

– descubrimiento de redes remotas,

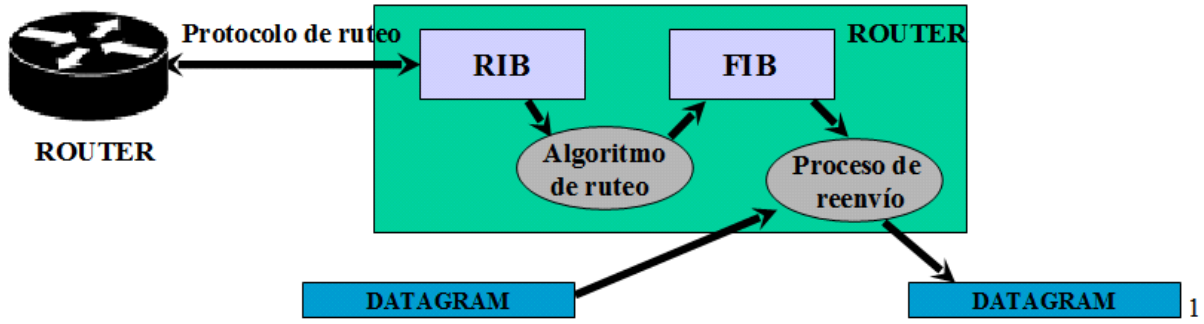
– mantenimiento de información de enrutamiento actualizada,

– selección de la mejor ruta hacia las redes de destino y

– capacidad de encontrar una mejor nueva ruta si la ruta actual deja de estar disponible.

Algoritmo de ruteo: Un algoritmo es una secuencia limitada de pasos que se usan para llevar a cabo una tarea. Los protocolos de enrutamiento usan algoritmos para facilitar información de enrutamiento y para determinar la mejor ruta.

Base de datos: información en cada router acerca de las rutas a los distintos destinos, con cierta(s) métrica(s).



Diferencia y relación entre la función de ruteo y la función de reenvío de paquetes

Algunas veces es útil distinguir entre el enrutamiento, que es el proceso consistente en tomar la decisión de cuáles rutas utilizar, y el reenvío, que consiste en la acción que se toma cuando llega un paquete. Se puede considerar que un enrutador realiza dos procesos internos. Uno de ellos maneja cada paquete conforme llega, buscando en las tablas de enrutamiento la línea de salida por la cual se enviará. Este proceso se conoce como reenvío. El otro proceso es responsable de llenar y actualizar las tablas de enrutamiento. Es ahí donde entra en acción el algoritmo de enrutamiento.

Ruteo

Envío de un paquete a su destino de la manera más eficiente posible.

Requiere un conocimiento global de la topología de la red.

Se adquiere a través de un protocolo de ruteo.

Reenvío

Proceso local en un router.

Cada paquete, se debe enviar lo más rápidamente al siguiente router camino al destino.

El reenvío se realiza en base a información provista por el componente de ruteo.

Cuáles son los recursos que utiliza la función de ruteo?. Explique de qué depende la cantidad de recursos de cada tipo utilizados. Mencione un protocolo que consuma una cantidad importante de recursos y otro que consuma muy pocos.

Memoria: tamaño de las tablas: RIB, FIB o de cualquier estructura utilizada por el algoritmo para el cálculo de ruta.

Tiempo de CPU: tiempo que se insume para realizar los cálculos necesarios para calcular las rutas.

Ancho de banda (intercambio de información entre routers): Cuánta carga adicional a los datos se envía en la red durante el ruteo.

Tiempo de administración (si requiere configuración manual).

Flooding: baja memoria, bajo tiempo de CPU, alto ancho de banda, bajo tiempo de administración.

Link state: alta memoria, alto tiempo de CPU, bajo ancho de banda, bajo tiempo de administración.

Explique las siguientes características que debería tener un algoritmo de ruteo: correctitud, simplicidad, robustez.

Correctitud: El algoritmo debe arrojar resultados correctos, los paquetes deben llegar a su destino sin problemas asociados al algoritmo.

Simplicidad: que no sea complejo.

Robustez: Una red puede tener que operar por años y experimentará fallas de software y hardware. El algoritmo de ruteo no debe requerir que se reinicialice la red después de fallas parciales.

Explique las siguientes características que debería tener un algoritmo de ruteo: estabilidad, equitatividad, optimalidad.

Estabilidad: el algoritmo debe llegar a un equilibrio y mantenerse en él.

Equitatividad: el algoritmo no debe “matar por inanición” a ningún host.

Optimización: debe ser eficiente, minimizar costos de espera.

Explique en qué consiste el ruteo estático, dinámico e híbrido. En qué casos se utiliza cada uno de ellos

Explicado anteriormente.

Estático: pocos hosts, o pocos enlaces, red estable, flujo poco variable.

Dinámico: redes inestables, con flujo variable.

Ruteo híbrido: ejemplo IGRP

Clasifique los protocolos estático, flooding, distancie vector, link state, path vector de acuerdo a que nodo o nodos toman las decisiones de ruteo, a la estrategia (cómo se adaptan a los cambios) de ruteo y al origen de la información que utilizan los nodos para tomar las decisiones de ruteo. Fundamente.

	nodos que toman las decisiones de ruteo	**estrategia de ruteo**	**proveniencia de datos para el ruteo**
Estático	distribuido	estático	local
Flooding	aislado	estático	local
Distance vector	distribuido	dinámico	adyacentes
Link state	distribuido	dinámico	global
Path vector	distribuido	dinámico	adyacentes

The goal is to find good paths to destination D. Each node advertises the path it prefers to get to D. For instance, when B gets two advertisements (CD and ED), it selects the one it prefers (CD) and advertises the path BCD to its neighbor.

The selection of preferred path follows a set of rules (e.g., avoid E because it is expensive or unreliable; prefer a shorter path, and so on.) Since the paths are advertised, the scheme avoids loops.

Explique el funcionamiento del ruteo centralizado. Ventajas y desventajas. Mejoras. Clasificación. Ejemplos de su uso.

Método de encaminamiento que implica uno o más centros de encaminamiento que recogen la información de tráfico de todos los nodos y generan actualizaciones de la tabla de encaminamiento para todos los nodos. Los algoritmos de encaminamiento centralizado van acompañados de un incremento de la sensibilidad al fallo, altos requisitos de procesamiento, y mayores cargas de las líneas de comunicaciones cerca del centro de encaminamiento debido a los datos entrantes y a las actualizaciones salientes. Un ejemplo de encaminamiento centralizado es TYMNET.

Periódicamente, cada IMP transmite la información de su estado al RCC. El RCC recoge toda esta información, y después, con base en el conocimiento total de la red completa, calcula las rutas óptimas de todo los IMP a cada uno de los IMP restantes, el encaminamiento centralizado también tiene algunos serios, si no es que

fatales, inconvenientes. La vulnerabilidad del RCC es un problema muy serio y para eso una solución es, tener una segunda máquina disponible como respaldo.

También se necesitará establecer un método de arbitraje para tener la seguridad de que el RCC primario y el de respaldo no lleguen a entrar en conflicto para saber quién es el jefe

Si el RCC calcula la ruta óptima para cada IMP, sin rutas alternas, la pérdida de tan solo una línea o IMP, llegará a desconectar algunos IMP del RCC, creando así terribles consecuencias para el sistema.

Si el RCC utiliza rutas alternas, se debilitara el argumento a favor de tener un RCC esto es el que pueda encontrar rutas óptimas.

Ventajas

- Sencillez (si son links estables)
- No se presentan inconsistencias entre las tablas de los distintos nodos.
- Los cálculos se realizan una sola vez, fuera de línea

Desventajas

- problemas ante cambios constantes en la red.
- La caída del nodo central podría generar la caída de toda la red.

Explique el funcionamiento del ruteo jerárquico. Ventajas y desventajas. Mejoras. Clasificación. Ejemplos de su uso.

Cuando se utiliza el enrutamiento jerárquico, los enrutadores se dividen en lo que llamaremos

regiones, donde cada enrutador conoce todos los detalles para enrutar paquetes a destinos dentro de su propia región, pero no sabe nada de la estructura interna de las otras regiones. Cuando se interconectan diferentes redes, es natural considerar cada una como región independiente a fin de liberar a los enrutadores de una red de la necesidad de conocer la estructura topológica de las demás.

Ventajas

- Tablas de ruteo significativamente más chicas.
- Transparencia entre regiones, o niveles de jerarquía.
- Ruteo más organizado.

Desventajas

- Muchas veces genera caminos más largos para los paquetes a enviar.

Explique el funcionamiento del ruteo utilizando flooding. Ventajas y desventajas. Mejoras. Clasificación. Ejemplos de su uso.

Cada paquete de entrada se envía por cada una de las líneas de salida, excepto aquella por la que llegó.

Mejoras

- Utilización del Spanning Tree
- No reenviar un paquete más de una vez (número de paquete)

- Contador de nodos y eliminación de paquete (TTL)
- Flooding selectivo

Ventajas

- Muy resistente a fallas
- Llega por el camino más corto
- Simple

Desventajas

- Genera gran overhead en la red

Usos

- Aplicaciones militares
- Bases de datos distribuidas

Explique las diferencias que considere más relevantes entre ruteo distancie vector y ruteo path vector.

En distancie vector los nodos envían a sus vecinos la información correspondiente a los costos de las rutas, en cambio en path vector cada nodo manda "el mejor camino" para llegar a otro nodo.

Al no tener costos, path vector evita el problema de conteo hasta el infinito.

Explique el funcionamiento de los protocolos tipo link state. Funciones de los routers.

El concepto en que se basa el enrutamiento por estado del enlace es sencillo y puede enunciarse en cinco partes. Cada enrutador debe:

1. Descubrir a sus vecinos y conocer sus direcciones de red.

    Mediante el envío de paquetes hello (request y reply) a sus vecinos.

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

    Mediante el envío de paquetes echo (request y reply).

3. Construir un paquete que indique todo lo que acaba de aprender.

    Con: emisor, número de secuencia, edad y vecinos. Costos

4. Enviar este paquete a todos los demás enrutadores.

    Mediante inundación con ciertas mejoras (edad y secuencia).

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

    Dijkstra.

Explique cómo funciona el ruteo para hosts móviles.

Habrá involucrados dos tipos de agentes

Agente base: es el encargado de administrar los host que habitualmente se encuentran en la red a la cual pertenece.

Agente foráneo: es el encargado de administrar la conexión de los host móviles nuevos que no pertenecen a su red.
En cada red deberían estar presentes ambos.

El ruteo para host móviles se podría resumir en los siguientes pasos:

1. Un host móvil registrado en un agente de base dado se mueve hacia una nueva red a la que no pertenece
2. Al ingresar allí, espera recibir un paquete del agente foráneo que le pida identificarse, este lo hace en broadcast a toda la red, o de no llegar le envía el mismo un paquete. Este paquete tendrá su dirección base, su propia dirección y un código de seguridad
3. Cuando el agente foráneo obtiene esa dirección, le envía al agente base de la red al cual pertenece el host móvil, un paquete que le indique la nueva ubicación del host móvil, con su correspondiente dirección y la clave de seguridad. El agente base comprueba la clave y el tiempo en el que fue enviado y a él le llego, y envía una confirmación.
4. Cuando un paquete es enviado con destino al host móvil, este es interceptado por el agente base, y reenviado a la nueva ubicación en la red a que se ubica en ese instante. Además se envía información al emisor del mensaje para que los nuevos paquetes sean enviados a la otra dirección.
5. Idealmente cuando el host móvil se desconecte debería avisarlo al agente foráneo para que este inicie el proceso inverso y el ruteo pueda estar como antes.

Qué son las redes MANET, cuáles son sus características principales. Mencione protocolos de ruteo adaptados a este tipo de redes.

Las redes manet son redes que se caracterizan por estar compuestas por nodos que involucran tanto un router como un host en una misma máquina. La topologías de estas redes pueden variar todo el tiempo y aun así deberían mantenerse activas. No necesitan de un protocolo estilo 802.11 para funcionar(WiFi).
La mayoría de estas redes se caracterizan por completar sus tablas de ruteo bajo demanda, es decir no intentan buscar más información a menos que necesiten enviar paquetes hacia direcciones que no conoce.

Un protocolo conocido que lo soporta es AODV ad hoc distancie vector.

Describa las distintas maneras de implementar ruteo broadcast en una red punto a punto (es decir, que no soporta él envió broadcast)

- Una primera alternativa seria envía paquetes distintos a todos los destinos en la red, pero implica que el emisor debería tener la lista completa de hosts. Además consumiría mucho ancho de banda.
- Otro método posible podría ser la inundación, pero sobrecarga demasiado la red.
- Otra forma es mediantes árbol de expansión, desde el router al cual pertenece el host que envía el mensaje, u otro árbol equivalente. Este método utiliza de manera óptima el ancho de banda, no obstante cada uno de los nodos debería conocer el árbol de expansión o contar con información para deducirlo. cada nodo reenviara los paquetes por cada línea perteneciente al árbol de expansión salvo por la que le llego. Distance vector no cuenta con información para algo de este estilo, pero links state si ya que contiene el mapa de red.
- Otro método es el de reenvío por ruta invertida:
    - El origen envía el paquete a todos sus vecinos
    - Cuando un nodo recibe un paquete chequea si este llega por el enlace correspondiente al "mejor camino" hasta el origen, de ser así lo reenvía a todos sus vecinos, de lo contrario no.

Explique en qué consisten los ruteos broadcast, multicast y anycast.

Los ruteos broadcast y multicast son estilos de ruteo donde un host desea envía un paquete a todo el resto de los hosts en la red o a varios de ellos al mismo tiempo respectivamente. En multicast se hace uso de una dirección especial la cual representara un conjunto de direcciones a las cual debe ser enviado el paquete. En anycast también se hace uso de una dirección similar, pero está en cambio indicara un conjunto de direcciones, y el paquete deberá llegar a ALGUNO de ellos. Estos pueden ser muy útil por ejemplo para acceder a información replicada.

Para ruteos broadcast y multicast es posible utilizar un spanning tree. Explique cómo se utiliza y las alternativas para construirlo.

El árbol de expansión es un subgrupo de la subred que incluye todos los enrutadores pero no contiene ciclos .De esta forma sabiendo el origen cada nodo reenviara el paquete por todas las rutas pertenecientes al spanning tree salvo por la ruta por la que le llego, logrando de esta manera optimizar el uso del canal para el transporte del paquete a todos sus destinos, sin caer en ciclos y reenvíos innecesarios de paquetes.

Alternativas anteriormente comentadas.

Explique la necesidad de crear grupos multicast. Mencione los tipos de grupos y describa las operaciones para el manejo de esos grupos.

Algunas aplicaciones requieren que procesos muy separados trabajen juntos en grupo; por

Ejemplo, un grupo de procesos que implementan un sistema de base de datos distribuido. En estos casos, con frecuencia es necesario que un proceso envíe un mensaje a todos los demás miembros del grupo. Si el grupo

es pequeño, simplemente se puede transmitir a cada uno de los miembros un mensaje punto a punto. Si el grupo es grande, esta estrategia es costosa.

Tipos de grupos

Operaciones para manejo de grupos: Se necesita alguna manera de crear y destruir grupos, y un mecanismo para que los procesos se unan a los grupos y salgan de ellos. La forma de realizar estas tareas no le concierne al algoritmo de enrutamiento. Lo que sí le concierne es que cuando un proceso se una a un grupo, informe a su host este hecho. Es importante que los enrutadores sepan cuáles de sus hosts pertenecen a qué grupos. Los hosts deben informar a sus enrutadores de los cambios en los miembros del grupo, o los enrutadores deben enviar de manera periódica la lista de sus hosts. De cualquier manera, los enrutadores aprenden qué hosts pertenecen a cuáles grupos. Los enrutadores les dicen a sus vecinos, de manera que la información se propaga a través de la subred.

Describa las características más relevantes del ruteo en la Internet.

- Escalable: crecimiento importante en cantidad de subredes y equipos interconectados.
- Adaptable a diferentes tecnologías de subred.
- Adaptable a cambios constantes en la topología (routers y conexiones) y en el tráfico.
- Posibilidad de configurar políticas (la red está dividida en diferentes administraciones, cada una con su propia política).
- Soporte de QoS.
- Jerárquico.
- Compatibilidad entre los diferentes protocolos de ruteo.

Dada la arquitectura de ruteo en la Internet, describa los tipos de ruteo (host-router, router-router, entre sistemas autónomos). ¿Cuál es el objetivo de cada uno de ellos y qué protocolos lo implementan?

Incompleto

Tipos de ruteo:

Host-router:

Router-router:

Entre sistemas autónomos:

Ruteo Link State - OSPF

Mencione los diferentes tipos de routers OSPF y explique sus funciones

OSPF distingue cuatro clases de enrutadores:

1. Enrutadores internos que están totalmente dentro de un área. Transfiere información entre routers adyacentes.

2. Enrutadores de límite de área que conectan dos o más áreas.

3. Enrutadores de la red dorsal que están en la red dorsal. Se comunican con otros routers para transferir información entre áreas..

4. Enrutadores fronterizos de sistemas autónomos que se comunican con los enrutadores de otros sistemas autónomos.

Mencione los distintos tipos de rutas OSPF y explique cuál es el significado de cada una de ellas.

Durante la operación normal, se pueden necesitar tres tipos de rutas: dentro del área, entre áreas y entre sistemas autónomos. Las rutas dentro del área son las más fáciles, puesto que el enrutador de origen ya conoce el camino más corto al enrutador de destino. El enrutamiento entre áreas siempre procede en tres pasos: va del origen a la red dorsal; va a través de la red dorsal al área de destino; va al destino. Este algoritmo fuerza una configuración de estrella en OSPF con la red dorsal actuando como concentrador y las otras áreas como rayos. Los paquetes se enrutan del origen al destino “como están”. No se encapsulan ni se entunelan, a menos que vayan a un área cuya única conexión a la red dorsal sea un túnel... PAGINA 457

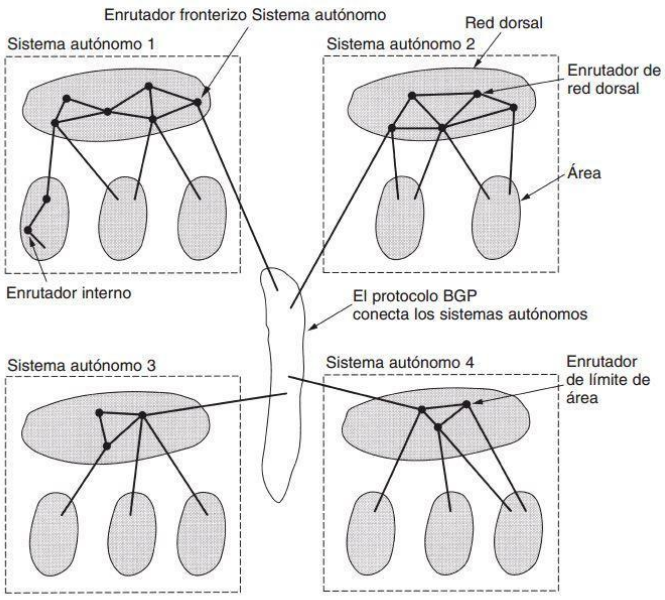


Figura 5-65. Relación entre sistemas autónomos, redes dorsales y áreas en OSPF.

Mencione y explique las características de los distintos tipos de áreas definidas en OSPF

Muchos de los sistemas autónomos en Internet son grandes por sí mismos y nada sencillos de administrar. OSPF les permite dividirlos en áreas numeradas donde un área es una red o un conjunto de redes inmediatas. Las áreas no se traslapan ni la necesidad es exhaustiva, es decir, algunos enrutadores pueden no pertenecer a área alguna.

Un área es una generalización de una subred. Fuera de un área, su topología y detalles no son visibles. Cada sistema autónomo tiene un área de red dorsal, llamada 0. Todas las áreas se conectan a la red dorsal, posiblemente por túneles, de modo que es posible entrar desde cualquier área en el sistema autónomo a cualquier otra área en el sistema autónomo mediante la red dorsal.

Tipos de áreas

- Backbone Área
  - Área principal
- Stub Área
  - Sólo conectada al backbone
  - No acepta rutas externas
  - Depende de default route para rutas externas
  - No acepta virtual links ni ASBRs
- Totally Stub Área
  - Clase de stub área
  - No propaga rutas
  - No acepta rutas inter área
  - Depende de default route para rutas externas e inter área
- Not-so-stubby Área
  - Clase de stub área
  - Acepta rutas externas (LSA 7) sólo para propagarlas dentro del SA

Area / Ruta	Default	Intra-área	Inter-área	Externa
No stub	Si	Si	Si	Si
Stub	Si	Si	Si	No
Totally stub	Si	Si	No	No
Not so stubby	Si	Si	Si	Si

¿Qué es un Designated Router (OSPF) y cuáles son sus funciones?

OSPF trabaja intercambiando información entre enrutadores adyacentes, que no es lo mismo que entre enrutadores vecinos. En particular, es ineficaz tener cualquier enrutador en la LAN que se comunica con cualquier otro enrutador en la LAN. Para evitar esta situación, se elige un enrutador como enrutador designado. Se dice que es adyacente a todos los demás enrutadores en su LAN, e intercambia información

con ellos. Los enrutadores vecinos que no son adyacentes no intercambian información entre sí. Un enrutador designado como respaldo siempre se guarda actualizado, para facilitar la transición en caso de que el primer enrutador designado se cayera y necesitara ser reemplazado de manera inmediata.

Explique cómo se elige un Designated Router (OSPF)

The DR is elected based on the following default criteria:

- If the priority setting on an OSPF router is set to 0, that means it can NEVER become a DR or BDR (Backup Designated Router).
- When a DR fails and the BDR takes over, there is another election to see who becomes the replacement BDR.
- The router sending the Hello packets with the highest priority wins the election.
- If two or more routers tie with the highest priority setting, the router sending the Hello with the highest RID (Router ID) wins. NOTE: a RID is the highest logical (loopback) IP address configured on a router, if no logical/loopback IP address is set then the Router uses the highest IP address configured on its active interfaces. (e.g. 192.168.0.1 would be higher than 10.1.1.2).
- Usually the router with the second highest priority number becomes the BDR.
- The priority values range between 0 - 255,[11] with a higher value increasing its chances of becoming DR or BDR.
- IF a HIGHER priority OSPF router comes online AFTER the election has taken place, it will not become DR or BDR until (at least) the DR and BDR fail.
- If the current DR 'goes down' the current BDR becomes the new DR and a new election takes place to find another BDR. If the new DR then 'goes down' and the original DR is now available, still previously chosen BDR will become DR.

Qué son los Link State Advertisments de OSPF?. Mencione los que conocen y explique para qué se utilizan.

- Anuncios realizados por los routers OSPF
- Contienen información topológica de varios tipos generada por el router que anuncia el LSA
- Se utilizan para permitir a los routers adquirir una noción exacta de la topología de la red
- Generación de LSAs
    - Periódicamente
    - Cuando se producen cambios en la topología
- Control de LSAs
    - Número de secuencia
    - Indicador de tiempo transcurrido desde que fueron generadas
- Difusión de LSAs
    - Flooding con asentimiento

TIPO	FUNCION
1-Router link advertisements	Conexiones de un router
2-Network link advertisements	Routers conectados a una red de acceso múltiple
3-ABR summary link advertisements	Costos a redes en áreas del SA
4-ASBR summary link advertisements	Costos a los ASBRs
5-AS external route advertisements	Costos a destinos externos (redes en otros SAs)
6-Multicast group LSA	Multicast
7-Not-so-stubby area (NSSAs) external	Codtos a destinos externos (otros SAs) anunciados dentro de un área stub

¿Cuáles son los tipos de subredes definidas por OSPF y en qué aspectos influyen sobre el protocolo?

- Tipos de subredes
  - Punto a punto
  - Multiacceso broadcast
  - Multiacceso no broadcast (NBMA)
  - Punto a multipunto
- Funciones afectadas
  - Descubrimiento y mantenimiento de neighbors
  - Sincronización de la base de datos topológica
  - Abstracción

Explique el proceso de reducción de tablas en OSPF

La reducción de tablas en OSPF se logra básicamente con dos mecanismos: la definición de distintas áreas (limitando las conexiones de los componentes internos) y el establecimiento de routers designados.

Explique cómo se calculan las rutas en OSPF (incluyendo área backbone)

Utilizando la inundación de mensajes, cada enrutador informa a todos los demás enrutadores en su área sobre sus vecinos y costos. Esta información permite a cada enrutador construir el grafo para su(s) área(s) y calcular la ruta más corta. El

Área de la red dorsal también hace esto. Además, los enrutadores de la red dorsal aceptan la información de los enrutadores del límite de área para calcular la mejor ruta de cada enrutador de la red dorsal a cada enrutador restante. Esta información se difunde a los enrutadores de límite de área que la anuncian dentro de sus áreas. Usando esta información, un enrutador que está a punto de enviar un paquete dentro del área puede seleccionar el enrutador de mejor salida a la red dorsal.

¿Cuál es la diferencia entre una relación de vecindad (neighbor) y una relación de adyacencia (adjacency) entre dos routers OSPF?

Todos los enrutadores en la misma LAN son vecinos. OSPF trabaja intercambiando información entre enrutadores adyacentes, que no es lo mismo que entre enrutadores vecinos. En particular, es ineficaz tener cualquier enrutador en la LAN que se comunica con cualquier otro enrutador en la LAN. Para evitar esta situación, se elige un enrutador como enrutador designado. Se dice que es adyacente a todos los demás enrutadores en su LAN, e intercambia información con ellos. Los enrutadores vecinos que no son adyacentes no intercambian información entre sí.

Ruteo distancie vector - RIP

Describa el problema de convergencia del ruteo Distance vector. Ejemplifique

El enrutamiento por vector de distancia funciona en teoría, pero tiene un problema serio en

La práctica: aunque llega a la respuesta correcta, podría hacerlo lentamente. En particular, reacciona con rapidez a las buenas noticias, pero con lentitud ante las malas.

Ahora consideremos la situación de la figura 5-10(b), en la que todas las líneas y enrutadores están activos inicialmente. Los enrutadores B, C, D y E tienen distancias a A de 1, 2, 3 y 4, respectivamente.

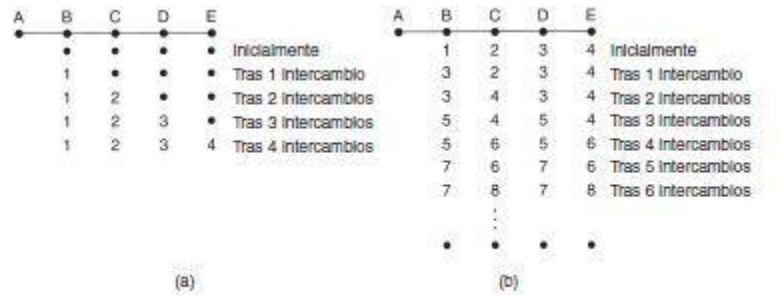


Figura 5-10. El problema de la cuenta hasta infinito.

Ante la caída de un router (en el ejemplo A) ningún enrutador jamás tiene un valor mayor en más de una unidad que el mínimo de todos sus vecinos. Gradualmente, todos los enrutadores elevan cuentas hacia el infinito, pero el número de intercambios requerido depende del valor numérico usado para el infinito. Por esta razón, es prudente hacer que el infinito sea igual a la ruta más larga, más 1.

¿Qué son los triggered updates de RIP?. En qué aspecto mejoran la performance del protocolo?

Las actualizaciones no se envían a los routers vecinos a menos que una de las cuatro cosas siguientes suceda:

- El router recibe una petición específica para una actualización.
- La información de otro interfaces ha causado una modificación de la base de datos RIP causando una actualización para ser activado.
- Una interfaz sube o baja, en cuyo caso se envía una actualización de base de datos parcial.
- Un router está encendido desencadena una actualización de la base de datos completa.

Esta técnica logra que se reduzca significativamente el ancho de banda invertido en mantener actualizadas las tablas del ruteo.

- Información de ruteo enviada asincrónicamente al producirse un cambio
- Aceleran tiempos de convergencia del algoritmo
- Complementan al intercambio periódico de tablas
- Pueden dar lugar a un volumen de tráfico considerable (dependiendo del estado de las tablas)

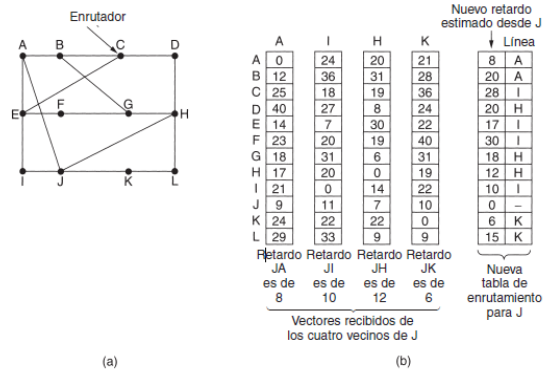
Describa la técnica de Split horizon. En qué aspecto mejora al protocolo?. Cuál es la mejora adicional si se utiliza poissonus reverse?. Ejemplifique

Se trata de una de las soluciones utilizadas para solventar el conteo a infinito. Es una modificación del algoritmo VD para evitar que un nodo informe a su vecino sobre la distancia que conoce hasta el nodo X cuando la trayectoria hacia X pasa a través de ese nodo vecino. El algoritmo por horizonte dividido consigue que las “malas noticias” se propaguen con la misma rapidez que las “buenas noticias”. Sin embargo este algoritmo no funciona para todas las combinaciones de topologías posibles por lo que sólo mitiga el problema sin solucionarlo.

Poisonous reverse Lo que realmente hace es informar que dicha distancia es infinita. De esta manera abarca aún más conflictos que Split horizon por si solo (técnica más agresiva, corta la ruta directamente).

Describa en detalle cómo un router procesa el vector de distancias recibido de un nodo vecino para obtener las rutas a las distintas redes. Ejemplifique

Este proceso de actualización se ilustra en la figura 5-9. En la parte (a) se muestra una subred. En las primeras cuatro columnas de la parte (b) aparecen los vectores de retardo recibidos de los vecinos del enrutador J. A indica tener un retardo de 12 mseg a B, un retardo de 25 mseg a C, un retardo de 40 mseg a D, etc. Suponga que J ha medido o estimado el retardo a sus vecinos A, I, H y K en 8, 10, 12 y 6 mseg, respectivamente. Considere la manera en que J calcula su nueva ruta al enrutador G. Sabe que puede llegar a A en 8, 10, 12 y 6 mseg, respectivamente



	A	I	H	K	Nuevo retardo estimado desde J	Línea
A	0	24	20	21	8	A
B	12	36	31	28	20	A
C	25	18	19	36	28	I
D	40	27	8	24	20	H
E	14	7	30	22	17	I
F	23	20	19	40	30	I
G	18	31	6	31	18	H
H	17	20	0	19	12	H
I	21	0	14	22	10	I
J	9	11	7	10	0	–
K	24	22	22	0	6	K
L	29	33	9	9	15	K

Figura 5-9. (a) Subred. (b) Entrada de A, I, H, K y la nueva tabla de enrutamiento de J.

Considere la manera en que J calcula su nueva ruta al enrutador G. Sabe que puede llegar a A en 8 mseg, y A indica ser capaz de llegar a G en 18 mseg, por lo que J sabe que puede contar con un retardo de 26 mseg a G si reenvía a través de A los paquetes destinados a G. Del mismo modo, J calcula el retardo a G a través de I, H y K en 41 (31 + 10), 18 (6 + 12) y 37 (31 + 6) mseg, respectivamente. El mejor de estos valores es el 18, por lo

que escribe una entrada en su tabla de enrutamiento indicando que el retardo a G es de 18 mseg, y que la ruta que se utilizará es vía H. Se lleva a cabo el mismo cálculo para los demás destinos, y la nueva tabla de enrutamiento se muestra en la última columna de la figura.

En definitiva se va a estar calculando la ruta y el coste, según los datos que me pasen los vecinos sumado al costo hace ese vecino dado

Explique en qué consisten el efecto de rebote (bouncing effect) y el conteo a infinito (count to infinity).

Anteriormente explicada.

Explique el intercambio de datos que realizan los routers cuando utilizan RIP (PDus intercambiadas, encapsulación, etc.).

INCOMPLETA

¿Cuáles son los timers utilizados por RIP y para qué se utilizan?

Timers

Intercambio de información de ruteo: 30 segundos

Validez de una ruta: 180 segs.

Garbage collection: 120 segs

Timer para triggered updates: al azar, entre 1 y 5 segs.

Congestión y calidad de servicio

Qué es la congestión a nivel de red? Explique cómo se produce. Cuál es la diferencia entre congestión y control de flujo?

Cuando hay demasiados paquetes presentes en la subred (o en una parte de ella), hay una degradación del desempeño. Esta situación se llama congestión.

Cuando la cantidad de paquetes descargados en la subred por los hosts está dentro de su capacidad de conducción, todos se entregan (excepto unos pocos afligidos por errores de transmisión) y la cantidad entregada es proporcional al número enviado. Sin embargo, a medida que aumenta el tráfico, los enrutadores ya no pueden manejarlo y comienzan a perder paquetes. Esto tiende a empeorar las cosas. Con mucho tráfico, el desempeño se desploma por completo y casi no hay entrega de paquetes.

La congestión puede ocurrir por varias razones.

- Si de manera repentina comienzan a llegar cadenas de paquetes por tres o cuatro líneas de entrada y todas necesitan la misma línea de salida, se generará una cola. Si no hay suficiente memoria para almacenar a todos los paquetes, algunos de ellos se perderán. La adición de memoria puede ayudar hasta cierto punto, pero Nagle (1987) descubrió que si los enrutadores tienen una cantidad infinita de memoria, la congestión empeora en lugar de mejorar, ya que para cuando los paquetes llegan al principio de la cola, su temporizador ha terminado (repetidamente) y se han enviado duplicados. Todos estos paquetes serán debidamente reenviados al siguiente enrutador, aumentando la carga en todo el camino hasta el destino.
- Los procesadores lentos también pueden causar congestión. Si las CPUs de los enrutadores son lentas para llevar a cabo las tareas de administración requeridas (búferes de encolamiento, actualización de tablas, etcétera), las colas pueden alargarse, aun cuando haya un exceso de capacidad de línea
- Las líneas de poco ancho de banda también pueden causar congestión.

Vale la pena indicar de manera explícita la diferencia entre el control de la congestión y el control de flujo, pues la relación es sutil. El control de congestión se ocupa de asegurar que la subred sea capaz de transportar el tráfico ofrecido. Es un asunto global, en el que interviene el comportamiento de todos los hosts, todos los enrutadores, el proceso de almacenamiento y

reenvío dentro de los enrutadores y todos los demás factores que tienden a disminuir la capacidad de transporte de la subred.

En contraste, el control de flujo se relaciona con el tráfico punto a punto entre un emisor dado y un receptor dado. Su tarea es asegurar que un emisor rápido no pueda transmitir datos de manera continua a una velocidad mayor que la que puede absorber el receptor. El control de flujo casi siempre implica una retroalimentación directa del receptor al emisor, para indicar al emisor cómo van las cosas en el otro lado.

Explique en términos generales en qué consisten las estrategias de prevención y las de detección y corrección de congestión. En qué casos utilizaría las primeras y en qué casos las segundas?. Mencione casos concretos.

Muchos problemas de los sistemas complejos, como las redes de computadoras, pueden analizarse desde el punto de vista de una teoría de control. Este método conduce a dividir en dos grupos todas las soluciones: de ciclo abierto (prevención) y de ciclo cerrado (detección y corrección). En esencia, las soluciones de ciclo abierto intentan resolver el problema mediante un buen diseño, para asegurarse en primer lugar de que no ocurra. Una vez que el sistema está en funcionamiento, no se hacen correcciones a medio camino.

Las herramientas para llevar a cabo control de ciclo abierto incluyen decidir cuándo aceptar

Tráfico nuevo, decidir cuándo descartar paquetes, y cuáles, y tomar decisiones de calendarización en varios puntos de la red. Todas tienen en común el hecho de que toman decisiones independientemente del estado actual de la red.

En contraste, las soluciones de ciclo cerrado se basan en el concepto de un ciclo de retroalimentación. Este método tiene tres partes cuando se aplica al control de congestión:

1. Monitorear el sistema para detectar cuándo y dónde ocurren congestiones.
2. Pasar esta información a lugares en los que puedan llevarse a cabo acciones.
3. Ajustar la operación del sistema para corregir el problema.

Explique las dos alternativas para la solución de estados de congestión. En qué casos aplicaría cada una de ellas?. Mencione casos concretos.

INCOMPLETA

Cuadro para las próximas tres preguntas.

Capa	Políticas
Transporte	• Política de retransmisión • Política de almacenamiento en caché de paquetes fuera de orden • Política de confirmaciones de recepción • Política de control de flujo • Determinación de terminaciones de temporizador
Red	• Circuitos virtuales vs. datagramas en la subred • Política de encolamiento y servicio de paquetes • Política de descarte de paquetes • Algoritmo de enrutamiento • Administración de tiempo de vida del paquete
Enlace de datos	• Política de retransmisiones • Política de almacenamiento en caché de paquetes fuera de orden • Política de confirmación de recepción • Política de control de flujo

Figura 5-26. Políticas relacionadas con la congestión.

Explique cada una de las medidas de prevención de congestión que pueden ser tomadas a nivel 2 (data link). Como influye cada una de ellas en la prevención de la congestión?

•**Política de retransmisiones**: se utiliza la repetición selectiva(1).

•**Política de almacenamiento en caché de paquetes fuera de orden**: con esto se logra disminuir la cantidad de retransmisiones. Cuando llega un paquete, aunque no haya llegado su anterior, se almacena en caché.

•**Política de confirmación de recepción**: no se responden de inmediato las recepciones de los paquetes. Si se espera para poder sobreponerlas en el tráfico inverso.

•**Política de control de flujo:** Un esquema de control de flujo estricto (por ejemplo, una ventana pequeña) reduce la tasa de datos y permite, por lo tanto, atacar la congestión.

(1)Repetición selectiva: se descarta una trama dañada recibida, pero las tramas en buen estado recibidas después de ésa se almacenan en el búfer. Cuando el emisor termina, sólo la última trama sin confirmación se retransmite.

Explique cada una de las medidas de prevención de congestión que pueden ser tomadas a nivel 3 (red). ¿Cómo influye cada una de ellas en la prevención de la congestión?

• **Circuitos virtuales vs. Datagramas en la subred:** muchos algoritmos de control de congestión sólo funcionan con subredes de circuitos virtuales.

• **Política de encolamiento y servicio de paquetes:** La política de encolamiento y servicio de paquetes se refiere a que los enrutadores tengan una cola por línea de entrada, y una o varias colas por línea de salida. También se relaciona con el orden en que se procesan los paquetes.

• **Política de descarte de paquetes:** La política de descarte es la regla que indica qué paquete se descarta cuando no hay espacio. Una buena política puede ayudar a aliviar la congestión y una mala puede hacerlo peor.

• **Algoritmo de enrutamiento:** Un buen algoritmo de enrutamiento puede evitar la congestión si distribuye el tráfico entre todas las líneas, pero uno malo puede enviar demasiado tráfico por líneas ya congestionadas.

• **Administración de tiempo de vida del paquete:** se encarga del tiempo que puede existir un paquete antes de ser descartado. Si este tiempo es demasiado grande, los paquetes perdidos pueden bloquear la operación durante un buen rato, pero si es demasiado corto, los paquetes pueden expirar antes de llegar a su destino, lo que provoca retransmisiones.

Explique cada una de las medidas de prevención de congestión que pueden ser tomadas a nivel 4 (transporte). ¿Cómo influye cada una de ellas en la prevención de la congestión?

• **Política de retransmisión:** IDEM NIVEL 2.

• **Política de almacenamiento en caché de paquetes fuera de orden:** IDEM NIVEL 2.

• **Política de confirmaciones de recepción:** IDEM NIVEL 2.

• **Política de control de flujo:** IDEM NIVEL 2.

• **Determinación de terminaciones de temporizador:** el tiempo de tránsito a

Través de la red es menos predecible que el tiempo de tránsito por un cable entre dos enrutadores. Por eso, si el intervalo es demasiado corto, se enviarán paquetes extra de manera

innecesaria. Si es muy largo, se reducirá la congestión, pero el tiempo de respuesta se verá afectado cada vez que se pierda un paquete.

De qué tipo son las medidas de control de congestión "control de admisión" y reserva de recursos?. Explique cada una de ellas. Se pueden utilizar en conjunto? en caso afirmativo explique cómo, en caso negativo justifique por qué.

Estas son medidas para el control dinámico de la congestión.

Control de admisión: una vez que se ha detectado la congestión, no se establecen circuitos virtuales nuevos hasta que ha desaparecido el problema. Es un método de simple implementación. Otra alternativa, es seguir estableciendo nuevos circuitos virtuales por rutas alternativas libres de problemas.

Reserva de recursos: se realiza un acuerdo entre el host y la subred cuando se establece un circuito virtual. Este arreglo normalmente especifica el volumen y la forma del tráfico, la calidad de servicio requerida y otros parámetros. Para cumplir con su parte del acuerdo, la subred por lo general reservará recursos a lo largo de la ruta cuando se establezca el circuito. Estos recursos pueden incluir espacio en tablas y en búfer en los enrutadores y ancho de banda en las líneas.

En caso de que exista la reserva de recursos no será necesario el control de admisión, ya que no habrá congestión si ya se tienen reservados los recursos a utilizar en toda la ruta (Ponele que la re invente yo, guarda).

Describa detalladamente la técnica de "feedback a los emisores" utilizada para detectar y corregir situaciones de congestión.

Hay dos grupos de estas técnicas, una de ellas es la de bit de advertencia, en donde

Un router, intermedio o no, modificara el paquete de asentimiento con el bit de advertencia en alto, indicando que la ruta esta congestionada. De esta manera el emisor baja la tasa de transmisión.

Observe que debido a que cada enrutador a lo largo de la ruta podía activar el bit de advertencia, el tráfico se incrementaba sólo cuando no había enrutadores con problemas.

Otra de las formas es enviar directamente un paquete de advertencia. Este paquete contendrá el destino, para que el emisor al recibirlo baje la tasa de transferencia hacia ese destino en particular.

Esta técnica puede aplicarse de dos maneras, enviando directamente al emisor para que al llegar se modifique el flujo, o ir modificando el flujo salto por salto. La ventaja de segundo es que el flujo se irá reduciendo y de esta manera la congestión disminuirá gradualmente, sin la necesidad de esperar hasta que el emisor reciba efectivamente el paquete.

Explique detalladamente la técnica de control de congestión ECN (Explicit Congestión Notification)

Es una extensión a IP y TCP que permite notificar de un host a otro, la congestión de la red sin descartar paquetes. Cuando ECN está establecido, el router que responde a ECN, marca la cabecera IP del paquete con el fin de anticipar una posible congestión. De esta manera se envía un hecho al emisor para que reduzca el flujo de paquetes.

Cuando decimos que ECN debe estar establecido, significa que se debe realizar la verificación de si los dos extremos soportan esta técnica. En caso que sí, se hace uso de la técnica.

Explique detalladamente la técnica de control de congestión RED (Random Early Detection)

Esta técnica se basa en la idea de que la mayoría de los paquetes perdidos en una red muy segura, como por ejemplo una cableada con bajo ruido, es la congestión producto del desborde de los buffers.

La idea subyacente será entonces descartar los paquetes antes de que la situación se vuelve irremediable. Para esto los routers conservan un promedio móvil del tamaño de sus colas. Como un router puede no saber cuál fue el host de origen que causo la congestión, entonces elegirá un paquete al azar de esa cola y lo eliminará.

El host origen al no recibir confirmación, y tratarse de uno que responde disminuyendo su velocidad ante la pérdida de paquete, bajará su tasa de transmisión. Obviamente si el host no usara es políticaica la técnica sería imposible de utilizar.

Explique detalladamente la técnica de control de congestión ICMP Source Quench

Cap. 6, completar

Mencione y explique algunas de las características de calidad de servicio requerida por las aplicaciones multimedia

Aplicación	Confiabilidad	Retardo	Fluctuación	Ancho de banda
Correo electrónico	Alta	Bajo	Baja	Bajo
Transferencia de archivos	Alta	Bajo	Baja	Medio
Acceso a Web	Alta	Medio	Baja	Medio
Inicio de sesión remoto	Alta	Medio	Media	Bajo
Audio bajo demanda	Baja	Bajo	Alta	Medio
Vídeo bajo demanda	Baja	Bajo	Alta	Alto
Telefonía	Baja	Alto	Alta	Bajo
Videoconferencia	Baja	Alto	Alta	Alto

Figura 5-30. Qué tan rigurosos son los requerimientos de calidad del servicio.

Confiabilidad:

Retardo

Fluctuación

Ancho de banda

Mencione 5 técnicas para mejorar la calidad de servicio y explique brevemente cada una.

Sobre aprovisionamiento

Esta técnica consiste en proporcionar la suficiente capacidad de enrutador, espacio en búfer y ancho de banda como para que los paquetes fluyan con facilidad. El problema con esta solución es que es costosa.

Almacenamiento en búfer (buffering)

Los flujos pueden almacenarse en el búfer en el lado receptor antes de ser entregados. Almacenarlos en el búfer no afecta la confiabilidad o el ancho de banda, e incrementa el retardo, pero atenúa la fluctuación.

Modelado de tráfico

El modelado de tráfico consiste en regular la tasa promedio (y las ráfagas) de la transmisión

De los datos. Cuando se establece una conexión, el usuario y la subred (es decir, el cliente y la empresa portadora) acuerdan cierto patrón de tráfico para ese circuito. Algunas veces esto se llama acuerdo de nivel de servicio. En tanto el cliente cumpla con su parte del contrato y sólo envíe los paquetes acordados, la empresa portadora promete entregarlos de manera oportuna. Tales acuerdos no son tan importantes para las transferencias de archivos pero sí para los datos en tiempo real.

Regulación de tráfico (cubeta con goteo)

De manera conceptual, cada host está conectado a la red mediante una interfaz que contiene una cubeta con goteo, es decir, una cola interna infinita. Si llega un paquete cuando la cola está llena, éste se descarta. El host puede poner en la red un paquete por pulso de reloj. Este mecanismo convierte un flujo desigual de paquetes de los procesos de usuario dentro del host en un flujo continuo de paquetes hacia la red, moderando las ráfagas y reduciendo en una buena medida las posibilidades de congestión.

Cubeta con goteo, con tokens

El algoritmo de cubeta con goteo impone un patrón de salida rígido a la tasa promedio, sin importar la cantidad de ráfagas que tenga el tráfico. En muchas aplicaciones es mejor permitir que la salida se acelere un poco cuando llegan ráfagas grandes, por lo que se necesita un algoritmo más flexible, de preferencia uno que nunca pierda datos. El algoritmo de cubeta con tokens es uno de tales algoritmos. En este algoritmo, la cubeta con goteo contiene tokens, generados por un reloj a razón de un token cada ΔTseg. Para que se transmita un paquete, éste debe capturar y destruir un token. El algoritmo de cubeta con tokens descarta los

Tokens (es decir, la capacidad de transmisión) cuando se llena la cubeta, pero nunca descarta

Los paquetes. En contraste, el algoritmo de cubeta con goteo descarta los paquetes cuando se llena la cubeta.

Reservación de recursos

Una vez que se tiene una ruta específica para una flujo, es posible reservar recursos a lo largo de esa ruta para asegurar que la capacidad necesaria esté disponible. Se pueden reservar tres tipos de recursos:

1. Ancho de banda.

2. Espacio de búfer.

3. Ciclos de CPU.

Control de admisión

Ahora nos encontramos en el punto en que el tráfico entrante de algún flujo está bien modelado y puede seguir una sola ruta cuya capacidad puede reservarse por adelantado en los enrutadores a lo largo de la ruta. Cuando un flujo de este tipo se ofrece a un enrutador, éste tiene que decidir, con base en su capacidad y en cuántos compromisos tiene con otros flujos, si lo admite o lo rechaza.

Debido a que muchas partes pueden estar involucradas en la negociación del flujo (el emisor, el receptor y todos los enrutadores a lo largo de la ruta), los flujos deben describirse de manera precisa en términos de parámetros específicos que se puedan negociar. Un conjunto de tales parámetros se conoce como especificación de flujo. Por lo general, el emisor (por ejemplo, el servidor de vídeo) produce una especificación de flujo que propone los parámetros que le gustaría utilizar. Conforme la especificación se

propague por la ruta, cada enrutador la examina y modifica los parámetros conforme sea necesario. Las modificaciones sólo pueden reducir el flujo, no incrementarlo (por ejemplo, una tasa más baja de datos, no una más grande). Cuando llega al otro extremo, se pueden establecer los parámetros.

Ruteo balanceado

La mayoría de los algoritmos de enrutamiento tratan de encontrar la mejor ruta para cada destino y envían a través de ella todo el tráfico a ese destino. Un método diferente que se ha propuesto para proporcionar una calidad de servicio más alta es dividir el tráfico para cada destino a través de diferentes rutas. Puesto que generalmente los enrutadores no tienen un panorama completo del tráfico de toda la red, la única forma factible de dividir el tráfico a través de múltiples rutas es utilizar la información disponible localmente. Un método simple es dividir el tráfico en fracciones iguales o en proporción a la capacidad de los enlaces salientes

Calendarización de paquetes

Si un enrutador maneja múltiples flujos, existe el peligro de que un flujo acapare mucha de su capacidad y limite a los otros flujos. El procesamiento de paquetes en el orden de arribo significa que un emisor agresivo puede acaparar la mayor parte de la capacidad de los enrutadores por los que pasan sus paquetes, lo que reduce la calidad del servicio para otros.

Distintos tipos de calendarización

- Encolamiento justo: los enrutadores tienen colas separadas para cada línea de salida, una por flujo. Cuando una línea se queda inactiva, el enrutador explora las diferentes colas de manera circular, y toma el primer paquete de la siguiente cola.
- Variante de encolamiento justo: explora las colas de manera repetida, byte por byte, hasta que encuentra el instante en el que finalizará cada paquete. A continuación, los paquetes se ordenan conforme a su tiempo de terminación y se envían en ese orden.
- Encolamiento justo ponderado: da más ancho de banda a los hosts con “más prioridad”

Explique detalladamente la técnica para mejorar la calidad de servicio “traffic shaping”

Cuando el host de origen obtiene el paquete regulador, se le pide que reduzca en un porcentaje X el tráfico enviado al destino especificado. Puesto que otros paquetes dirigidos al mismo destino probablemente ya están en camino y generarán más paquetes reguladores, el host debe ignorar los paquetes reguladores que se refieran a ese destino por un intervalo fijo de tiempo. Una vez que haya expirado ese tiempo, el host escucha más paquetes reguladores durante otro intervalo. Si llega alguno, la línea todavía está congestionada, por lo que el host reduce el flujo aún más y comienza a ignorar nuevamente los paquetes reguladores. Si no llega ningún paquete de este tipo durante el periodo de escucha, el host puede incrementar el flujo otra vez. La retroalimentación implícita de este protocolo puede ayudar a evitar la congestión que aún no estrangula ningún flujo a menos que ocurra un problema.

Explique detalladamente la técnica para mejorar la calidad de servicio "buffering"

Los flujos pueden almacenarse en el búfer en el lado receptor antes de ser entregados. Almacenarlos en el búfer no afecta la confiabilidad o el ancho de banda, e incrementa el retardo, pero atenúa la fluctuación. Para el vídeo o audio bajo demanda, la fluctuación es el problema principal, por lo tanto, esta técnica es muy útil.

El problema que tiene esta técnica es que si al momento de consumir un paquete, este no se encuentra aún en el buffer, se va a producir un hueco por un tiempo. La manera de solucionarlo sería retrasar más el comienzo de la reproducción pero el problema seguiría latente.

Explique detalladamente la técnica para mejorar la calidad de servicio "packet scheduling"

La técnica de packet scheduling se basa en la idea de programar él envió de los diferentes paquetes, teniendo en cuenta que calidad de servicio se requiere. El objetivo principal de esta técnica es maximizar la capacidad del sistema, mientras satisface las QoS del usuario, alcanzando así cierto nivel de equilibrio. La mayoría de los algoritmos referidos a esta técnica, sirven para alcanzar: eficiencia, protección, flexibilidad y baja complejidad.

Existen diferentes algoritmos para él envió de paquetes:

Wireline

- FCFS (First-Come-First-Served)
- Round-Robin
- Strict priority: Este algoritmo asigna clases a las diferentes transmisiones, atacando estos diferentes niveles de QoS, y teniendo prioridades. Se lo llama strict porque no resuelve el problema de inanición.
- EDF(Earliest Deadline First)
- GPS(Generalized Processor Sharing)
- PGPS(Packet-by-packet GPS)

Wireless

- Idealized Wireless Fair Queueing (IWFQ)
- Channel-condition Independent packet Fair Queueing (CIF-Q)
- Improved Channel State Dependent Packet Scheduling (I-CSDPS)
- Proportional Fair

Mencione, explique y compare los frameworks para calidad de servicio propuestos en el ámbito de la IRTF

RSVP—Protocolo de reservación de recursos: se reservan los recursos necesarios para realizar la comunicación. Se reutilizan los caminos ya calculados y los recursos de los mismos.

Servicios diferenciados o basados en clase

En esta técnica la forma de transmisión de paquetes variara según la clase de servicios que se haya acordado. Por ejemplo un servicio que requiera muy pocos fluctuaciones en el flujo de datos será prestado mediante la técnica de la cubeta, mientras que un servicio que involucre reproducción de video podría utilizar podría usarse buffering.

Reenvío acelerado(o expedito):

Existen dos clases de servicios, expedito y regular. Al primero se le da más ancho de banda, proporcional al promedio histórico de paquetes de ese tipo. Se aplica un round robin ponderado dándole más prioridad a estos.

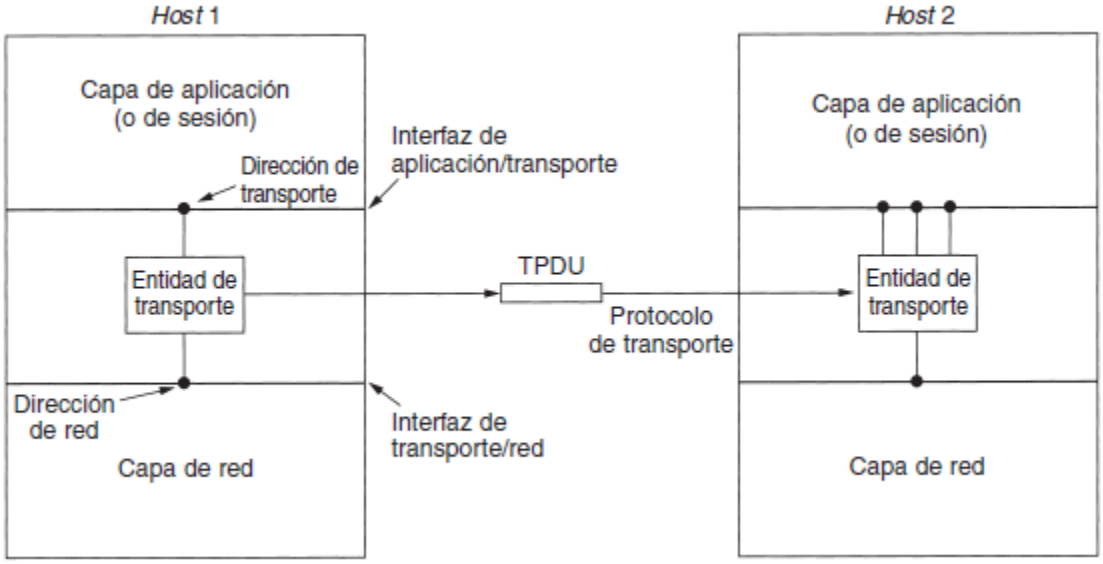
Reenvío asegurado

Especifica que deberán haber cuatro clases de prioridades, y cada una tendrá sus propios recursos. Además, define tres probabilidades de descarte para paquetes que están en congestión: baja, media y alta. En conjunto, estos dos factores definen 12 clases de servicios. El paso 1 es clasificar los paquetes en una de cuatro clases de prioridades. El paso 2 es marcar los paquetes de acuerdo con su clase. El paso 3 es pasar los paquetes a través de un filtro modelador/eliminador que podría retardar o descartar algunos de ellos, si hay muchos paquetes, algunos de ellos podrían descartarse aquí, mediante una categoría de eliminación.

Nivel de transporte – TCP, UDP

Explique cuál es la importancia del nivel de transporte en la arquitectura TCP/IP

La meta fundamental de la capa de transporte es proporcionar un servicio eficiente, confiable y económico a sus usuarios, que normalmente son procesos de la capa de aplicación. Para lograr este objetivo, la capa de transporte utiliza los servicios proporcionados por la capa de red. El hardware o software de la capa de transporte que se encarga del trabajo se llama entidad de transporte, la cual puede estar en el kernel (núcleo) del sistema operativo, en un proceso de usuario independiente, en un paquete de biblioteca que forma parte de las aplicaciones de red o en la tarjeta de red.



El código de transporte se ejecuta por completo en las máquinas de los usuarios, pero la capa de red, por lo general, se ejecuta en los enrutadores, los cuales son operados por la empresa portadora. La existencia de la capa de transporte hace posible que el servicio de transporte sea más confiable que el servicio de red subyacente. La capa de transporte puede detectar y compensar paquetes perdidos y datos alterados. Gracias a la capa de transporte, es posible escribir programas de aplicación usando un conjunto estándar de primitivas, y que estos programas funcionen en una amplia variedad de redes sin necesidad de preocuparse por lidiar con diferentes interfaces de subred y transmisiones no confiables.

Esta capa cumple la función clave de aislar a las capas superiores de la tecnología, el diseño y las imperfecciones de la subred.

El nivel de transporte y el nivel 2 se caracterizan por incluir funciones similares, ya que en ambos casos permiten la comunicación de 2 procesos remotos. Explique por qué las mismas funciones que en el nivel 2 son relativamente simples se hacen más complejas en el nivel 4. Dé ejemplos de al menos tres de ellas.

Existen diferencias significativas entre los dos, las cuales se deben a diferencias importantes entre los entornos en que operan ambos protocolos, como se muestra en la figura. En la capa de enlace de datos, dos

enrutadores se comunican directamente mediante un canal físico mientras que, en la capa de transporte, ese canal físico es reemplazado por la subred completa.

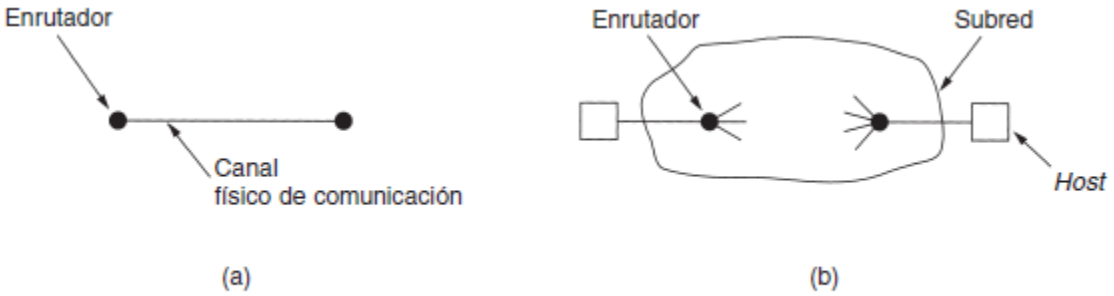


Figura 6-7. (a) Entorno de la capa de enlace de datos. (b) Entorno de la capa de transporte.

Las diferencias

1. Especificar el destino con el que se quiere comunicar. Nivel 2: no lo específica, única salida. Nivel 4: debe especificarlo.
2. Establecimiento de conexión: Nivel 2: innecesario, el otro extremo siempre está ahí. Nivel 4: se debe establecer la comunicación inicialmente.
3. Existencia potencial de capacidad de almacenamiento en subred.
4. Cantidad de buffers y control de flujo: Nivel 2: protocolos para que siempre que llegue una trama haya un buffer disponible. Nivel 4: protocolos más complejos.

Mencione y explique brevemente al menos 6 de las funciones del nivel de transporte.

Direccionamiento

Cuando un proceso desea establecer una conexión con un computador de aplicación remoto, debe especificar a cuál se conectará (¿a quién le mensaje?). El método que normalmente se emplea es definir direcciones de transporte en las que los procesos pueden estar a la escucha de solicitudes de conexiones. En Internet, estos puntos terminales se denominan puertos, pero usaremos el término genérico de TSAP (Punto de Acceso al Servicio de Transporte). Los puntos terminales análogos de la capa de red se llaman NSAP (Punto de Acceso al Servicio de Red). Las direcciones IP son ejemplos de NSAPS.

Establecimiento de la comunicación

El establecimiento de una conexión parece fácil, pero en realidad es sorprendentemente difícil. A primera vista, parecería que es suficiente con mandar una TPDU (Unidad de Datos del Protocolo de Transporte) con la petición de conexión y esperar a que el otro acepte la conexión. El problema viene cuando la red puede perder, almacenar, o duplicar paquetes. El principal problema es la existencia de duplicados retrasados. Esto puede solucionarse de varias maneras (ninguna es muy satisfactoria). Una es utilizar direcciones de transporte desechables. En este enfoque cada vez que necesitemos una dirección la creamos. Al liberarse la conexión descartamos la dirección y no se vuelve a utilizar. O también asignar una secuencia dentro de los

datos transmitidos, pero estos plantean el problema de que si se pierde la conexión perdemos el orden del identificador y ya no funciona. La solución sería más fácil si los paquetes viejos se eliminaran de la subred cada cierto tiempo de vida. Para ello podemos utilizar las siguientes técnicas: Un diseño de subred Restringido. Colocar un contador de saltos en cada paquete. Marcar el tiempo de cada paquete. Pero en la práctica no vale solo con hacer esto sino que tenemos que garantizar que todas las confirmaciones de los paquetes también se eliminan.

Liberación de la conexión

La liberación de una conexión es más fácil que su establecimiento. No obstante, hay más escollos de los que uno podría imaginar. Hay dos estilos de terminación de una conexión: liberación asimétrica y liberación simétrica. La liberación asimétrica es la manera en que funciona el mecanismo telefónico: cuando una parte cuelga, se interrumpe la conexión. La liberación simétrica trata la conexión como dos conexiones unidireccionales distintas, y requiere que cada una se libere por separado. La liberación asimétrica es abrupta y puede resultar en la perdida de datos. Por lo que es obvio que se requiere un protocolo de liberación más refinado para evitar la pérdida de datos. Una posibilidad es usar la liberación simétrica, en la que cada dirección se libera independientemente de la otra. Aquí, un host puede continuar recibiendo datos aun tras haber enviado una TPDU de desconexión.

La liberación simétrica es ideal cuando un proceso tiene una cantidad fija de datos por enviar y sabe con certidumbre cuándo los ha enviado. En otras situaciones, la determinación de si se ha efectuado o no todo el trabajo y se debe terminarse o no la conexión no es tan obvia. Podríamos pensar en un protocolo en el que el host 1 diga: “Ya termine, ¿Terminaste también?”. Si el host 2 responde “Ya termine también. Adiós”, la conexión puede liberarse con seguridad.

Pero no es tan fiable por el problema de que siempre tendremos que esperar la confirmación de los mensajes recibidos y si esta confirmación no llega no libera la conexión y después puede que necesite la confirmación de que llego la confirmación y entraríamos en un bucle del que no podemos salir.

Podemos hacer que al host 1 si no le llega la confirmación después de N intentos (es que quiere la desconexión), se libere. Esto produce una conexión semiabierta en la que el host 1 está desconectado pero el host 2 no como no le llega la confirmación no se desconecta nunca. Para solucionar esto creamos una regla por la cual si al host 2 no le llega ninguna TPDU durante cierta cantidad de segundos, se libera automáticamente.

Control de flujo y almacenamiento en buffers

Respecto de la manera en que se manejan las conexiones mientras están en uso, uno de los aspectos clave es el control de flujo. Se necesita un esquema para evitar que un emisor rápido desborde a un receptor lento. La diferencia principal es que un enrutador por lo regular tiene relativamente pocas líneas, y un host puede tener numerosas conexiones. Esta diferencia hace poco práctico emplear la implementación que se hace en la capa de enlace.

En esta capa lo que se hace es que si el servicio de red no es confiable, el emisor debe almacenar en un buffer todas las TPDUs enviadas, igual que en la capa enlace de datos. Sin embargo, con un servicio de red confiable

son posibles otros arreglos. En particular, si el emisor sabe que el receptor siempre tiene espacio de buffer, no necesita tener copias de las TPDUs que envía. Sin embargo, si el receptor no garantiza que se aceptará cada TPDU que llegue, el emisor tendrá que usar buffers de todas maneras. En el último caso, el emisor no puede confiar en la confirmación de recepción de la capa red porque esto sólo significa que ha llegado la TPDU, no que ha sido aceptada.

Los Buffers pueden ser de tres tipos, y usaremos cada uno de ellos cuando más nos convenga.

El equilibrio óptimo entre el almacenamiento del buffer en el origen y en el destino depende del tipo de tráfico transportado por la conexión.

Multiplexión

La multiplexión de varias conversaciones en conexiones, circuitos virtuales o enlaces físicos desempeña un papel importante en diferentes capas de la arquitectura de red. En la capa de transporte puede surgir la necesidad de multiplexión por varias razones. Por ejemplo, si en un host sólo se dispone de una dirección de red, todas las conexiones de transporte de esa máquina tendrán que utilizarla. Cuando llega una TPDU, se necesita algún mecanismo para saber a cuál proceso asignarla. Esta situación se conoce como multiplexión hacia arriba.

La multiplexión también puede ser útil en la capa transporte para la utilización de circuitos virtuales, que dan más ancho de banda cuando se reasigna a cada circuito una tasa máxima de datos. La solución es abrir múltiples conexiones de red y distribuir el tráfico entre ellas. Esto se denomina multiplexión hacia abajo.

Recuperación de caídas

Si los hosts y los enrutadores están sujetos a caídas, la recuperación es fundamental. Si la entidad de transporte está por entero dentro de los hosts, la recuperación de caídas de red y de enrutadores es sencilla. Si la capa de red proporciona servicio de datagramas, las entidades de transporte esperan pérdida de algunas TPDUs todo el tiempo, y saben cómo manejarla. Si la capa de red proporciona servicio orientado a la conexión, entonces la pérdida de un circuito virtual se maneja estableciendo otro nuevo y sondeando la entidad de transporte remota para saber cuáles TPDUs ha recibido y cuáles no.

Un problema más complicado es la manera de recuperarse de caídas del host. Al reactivarse, sus tablas están en el estado inicial y no sabe con precisión donde estaba.

En un intento por recuperar su estado previo, el servidor podría enviar una TPDU de difusión a todos los demás host, anunciando que se acaba de caer y solicitando a todos sus clientes que le informen el estado de todas la conexiones abiertas.

Explique en detalle las siguientes funciones del nivel de transporte:

Control de secuencia de las unidades de transmisión del nivel transporte (T-PDUs), Segmentación de las unidades de servicio a nivel transporte (T-SDUs) en unidades de transmisión de nivel transporte (T-PDUs), recuperación de errores.

Para cada una de ellas, explique si se encuentra presente en los protocolos TCP y/o UDP de qué manera.

INCOMPLETA

Control de secuencia de las unidades de transmisión del nivel de transporte (T-PDUs):

Los T-PDUs son los mensajes enviados de una entidad de transporte a otra. A continuación se presenta un ejemplo de alguna de ellas.

Primitiva	Paquete enviado	Significado
LISTEN	(ninguno)	Se bloquea hasta que algún proceso intenta la conexión
CONNECT	CONNECTION REQ.	Intenta activamente establecer una conexión
SEND	DATA	Envía información
RECEIVE	(ninguno)	Se bloquea hasta que llega un paquete DATA
DISCONNECT	DISCONNECTION REQ.	Este lado quiere liberar la conexión

Figura 6-2. Primitivas de un servicio de transporte sencillo.

En TCP

Segmentación de las unidades de servicio a nivel transporte (T-SDUs) en unidades de transmisión de nivel transporte (T-PDUs),

Recuperación de errores

Explique en detalle las siguientes funciones del nivel de transporte:

a) **Blocking/unblocking (agrupar/desagrupar) de varias unidades de servicio a nivel transporte (T-SDUs) en una unidad de transmisión del nivel transporte (T-PDU).**

b) **Detección de errores.**

c) **Control de flujo.**

Para cada una de ellas, explique si se encuentra presente en la arquitectura TCP/IP y de qué manera.

Leer en

http://www.cs.utexas.edu/users/chris/nph/CYCLADES/Iam/Cyclades/Tour/Transport%20Layer/Transport%20Layer.htm

Explique en detalle las siguientes de qué manera el nivel de transporte de la arquitectura TCP/IP colabora para evitar la congestión de la red (a nivel IP).

Al establecerse una conexión, se tiene que seleccionar un tamaño de ventana adecuado. El receptor puede especificar una ventana con base en su tamaño de búfer. Si el emisor se ajusta a su tamaño de ventana, no ocurrirán problemas por desbordamiento de búferes en la terminal receptora, pero aún pueden ocurrir debido a congestión interna en la red.

La solución de Internet es aceptar que existen dos problemas potenciales (capacidad de la red y capacidad del receptor) y manejarlos por separado. Para ello, cada emisor mantiene dos ventanas: la ventana que ha otorgado el receptor y una segunda ventana, la ventana de congestión. Cada una refleja la cantidad de bytes que puede enviar el emisor. La cantidad de bytes que pueden enviarse es la cifra menor de las dos ventanas.

Arranque lento: La ventana de congestión sigue creciendo exponencialmente hasta ocurrir una expiración del temporizador o alcanzar el tamaño de la ventana receptora.

(Ver si es a nivel IP).

Qué son los ports ofrecidos a las aplicaciones por las entidades de nivel transporte (TCP, UDP, etc.)? Cómo se asignan y cuál es su utilidad?. Qué son los "wellknown" ports?

Los puertos ofrecidos por las entidades de nivel de transporte son llamados sockets. Cada socket tiene un número (dirección), que consiste en la dirección IP del host, y un número de 16 bits, que es local a ese host, llamado puerto. Un puerto es el nombre TCP para un TSAP. Para obtener el servicio TCP, se debe establecer de manera explícita una conexión entre un socket en la máquina emisora y uno en la máquina receptora.

Un socket puede utilizarse para múltiples conexiones al mismo tiempo. En otras palabras, dos o más conexiones pueden terminar en el mismo socket. Las conexiones se identifican mediante los identificadores de socket de los dos extremos, es decir (socket1, socket2). No se utiliza ningún otro número de circuitos virtuales ni identificador.

"Wellknown" ports

Los números de puerto menores que 1024 se llaman puertos bien conocidos y se reservan para servicios estándar. Por ejemplo, cualquier proceso que desee establecer una conexión a un host

para transferir un archivo utilizando FTP puede conectarse con el puerto 21 del host de destino para conectar su demonio (daemon) FTP.

Asignación dinámica de ports: describa en detalle cómo funciona el mapeo de ports (portmapper). Cuál es su utilidad. Casos de uso.

El port mapper es un servicio ONC RPC (Open Network Computing Remote Procedure Call) que corre sobre los nodos de redes que proveen otros servicios ONC RPC.

La versión 2 del protocolo mapea el par numero/versión de un programa con un puerto. Cuando un servidor ONC RPC se inicia, le comunica al port mapper que numero de puerto está utilizando para cada par numero/versión que soporta (para un protocolo de transporte particular).

Los clientes que desean llamar a algún servicio ONC RPC deben primero contactarse con el port mapper del servidor.

El servicio port mapper siempre usa el puerto TCP o UDP 111 (debe haber un puerto acordado para que el cliente se pueda comunicar). El port mapper debe iniciarse antes que cualquier otro servidor RPC.

Describa el proceso de demultiplexing a que es sometido un frame Ethernet que es recibido por un host. Explique en base a qué campos las distintas entidades del host (driver, IP, etc.) determinan a qué entidad o proceso de nivel superior entregar la información que desencapsula. Dé un ejemplo concreto para un dato que está dirigido a un servidor web.

RTP: Real-time Transport Protocol.

El encabezado RTP consiste de tres palabras de 32 bits más algunas extensiones. La primera es el campo versión. El bit P indica que el paquete se ha rellenado a un múltiplo de 4 bytes. El último byte de relleno indica cuántos bytes se agregaron. El bit X indica que hay un encabezado de extensión. El formato y el significado de este encabezado no se definen. Lo único que se define es que la primera palabra de la extensión proporciona la longitud. El campo CC indica cuántos orígenes de contribución están presentes, de 0 a 15. El bit M es un bit marcador específico de la aplicación. Puede utilizarse para marcar el inicio de un cuadro de vídeo, el inicio de una palabra en un canal de audio, o algo más que la aplicación entienda. El campo Tipo de carga útil indica cuál algoritmo de codificación se ha utilizado. Puesto que cada paquete lleva este campo, la codificación puede cambiar durante la transmisión. El Número de secuencia es simplemente un contador que se incrementa en cada paquete RTP enviado. Se utiliza para detectar paquetes perdidos. El origen del flujo produce la marca de tiempo para indicar cuándo se creó la primera muestra en el paquete. Este valor puede ayudar a reducir la fluctuación en el receptor al desacoplar la reproducción del tiempo de llegada del paquete. El Identificador de origen de sincronización indica a cuál flujo pertenece el paquete. Es el método utilizado para multiplexar y desmultiplexar varios flujos de datos en un solo flujo de paquetes UDP. Por último, los Identificadores de origen de contribución, en caso de que haya, se utilizan cuando los mezcladores están presentes en el estudio. En ese caso, el mezclador es el origen de sincronización, y los flujos que se mezclan se listan aquí.

Byte 0				Byte 1		Byte 2	Byte 3
V	P	X	CC	M	PT	Sequence Number	
Time Stamp							
Synchronization Source (SSRC)							
Content Source (CSRC)							
Extension header (EH - opcional)							
Datos							

Describa brevemente el servicio que ofrecen los siguientes protocolos de nivel transporte: TCP, UDP, DCCP, SCTP.

TCP

Se diseñó específicamente para proporcionar un flujo de bytes confiable de extremo a extremo a través de una interred no confiable. La capa IP no proporciona ninguna garantía de que los datagramas se entregarán de manera apropiada, por lo que corresponde a TCP terminar los temporizadores y retransmitir los datagramas

conforme sea necesario. Los datagramas que llegan podrían hacerlo en el orden incorrecto; también corresponde a TCP reensamblarlos en mensajes en la secuencia apropiada. En resumen, TCP debe proporcionar la confiabilidad que la mayoría de los usuarios desean y que IP no proporciona.

UDP

Este protocolo proporciona una forma para que las aplicaciones envíen datagramas IP encapsulados sin tener que establecer una conexión. En el encabezado, posee los dos puertos, de origen y destino. Cuando llega un paquete UDP, su carga útil se entrega al proceso que está enlazado al puerto de destino. Un área en la que UDP es especialmente útil es en las situaciones cliente-servidor. Con frecuencia, el cliente envía una solicitud corta al servidor y espera una respuesta corta. Si se pierde la solicitud o la respuesta, el cliente simplemente puede terminar y probar nuevamente. El código no sólo es simple, sino que se necesitan muy pocos mensajes (uno en cada dirección) en comparación con un protocolo que requiere una configuración inicial. Puede enviar al servidor DNS un paquete UDP que contenga el nombre de dicho host. El servidor responde con un paquete UDP que contiene la dirección IP del host.

DCCP (Datagram Congestion Control Protocol):

Está pensado para aplicaciones que requieren la semántica basada en flujo de TCP pero no necesitan la entrega en orden ni la confiabilidad que ofrece TCP, o que quieren un control de congestión dinámico distinto del de TCP. De igual modo, DCCP está formulado para aplicaciones que no requieren de las características especiales de SCTP, como por ejemplo la entrega secuencial de flujo múltiple. Entre las aplicaciones que querrían implementar el DCCP se incluyen aquellas que tienen necesidad de entrega rápida de datos, y que combinada con la implementación de algún método de control de congestión, resulta generalmente en el arribo extemporáneo de la información, convirtiéndose en inútil.

SCTP (Stream Control Transmission Protocol)

Capacidad de Multihoming, en la cual uno (o dos) de los extremos de una asociación (conexión) pueden tener más de una dirección IP. Esto permite reaccionar en forma transparente ante fallos en la red. Entrega de los datos en trozos que forman parte de flujos independientes y paralelos eliminando así el problema de head of

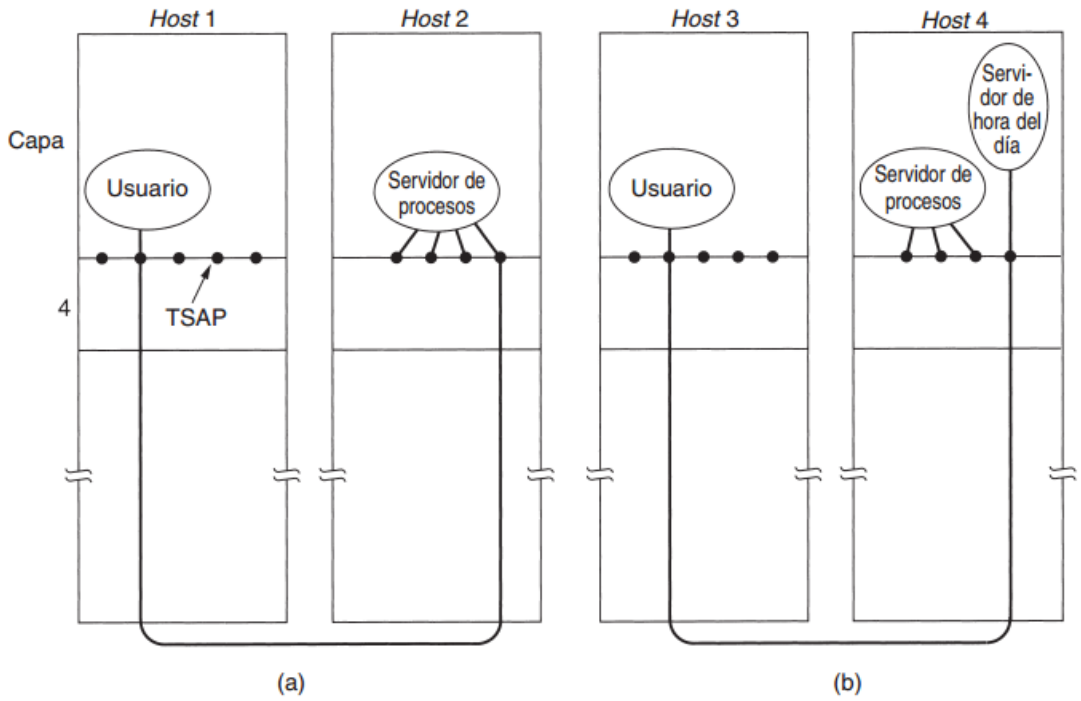
the line blocking que sufre TCP. Es capaz de seleccionar y monitorizar caminos, seleccionando un camino "primario" y verificando constantemente la conectividad de cada uno de los caminos alternativos. Mecanismos de validación y asentimiento como protección ante ataques por inundación, proveyendo notificación de trozos de datos duplicados o perdidos.

Para qué se utiliza y qué es un server de procesos? (por ejemplo el inetdaemon – inetd- de Linux).

Actúa como proxy de los servidores de menor uso. Este servidor escucha en un grupo de puertos al mismo tiempo, esperando una solicitud de conexión. Los usuarios potenciales de un servicio comienzan por emitir una solicitud CONNECT, especificando la dirección TSAP del servicio que desean.

Si no hay ningún servidor esperándolos, consiguen una conexión al servidor de procesos. Tras obtener la solicitud entrante, el servidor de procesos genera el servidor solicitado, permitiéndole heredar la conexión con el usuario existente. El nuevo servidor entonces hace el trabajo requerido, mientras que el servidor de procesos retorna a escuchar solicitudes nuevas.

Aunque el protocolo de conexión inicial funciona bien para aquellos servidores que pueden crearse conforme son necesarios, hay muchas situaciones en las que los servicios existen independientemente del servidor de procesos. Por ejemplo, un servidor de archivos necesita operar en un hardware especial (una máquina con un disco) y no puede simplemente crearse sobre la marcha cuando alguien quiere comunicarse con él.



Inetd es un deamon presente en la mayoría de sistemas tipo Unix, conocido como el "Súper Servidor de Internet", ya que gestiona las conexiones de varios demonios. La ejecución de una única instancia de inetd

reduce la carga del sistema, en comparación con lo que significaría ejecutar cada uno de los deamons que gestiona, de forma individual.

Inetd es un deamon que atiende las solicitudes de conexión que llegan a nuestro equipo, y está a la espera de todos los intentos de conexión que se realicen en una máquina. Cuando le llega una solicitud de conexión, irá dirigida a un puerto, por ejemplo, el 80 sería una solicitud al servidor de páginas web, 23 para telnet, 25 para SMTP, etc.

Inetd se utiliza principalmente para lanzar procesos que albergan a otros deamons, pero inetd también se utiliza para gestionar determinados protocolos triviales como chargen,auth y daytime.

Mencione y explique brevemente cinco características relevantes de UDP

1. No orientado a la conexión.
2. Sin control de flujo.
3. Sin control de congestión.
4. Sin control de errores.
5. Demultiplexa varios procesos en distintos puertos.

Mencione y explique brevemente cinco características relevantes de TCP

1. Orientado a la conexión.
2. Con control de flujo.
3. Con control de Congestion.
4. Con control de errores.
5. Todas las conexiones son duplex.

Mencione los campos más importantes de los frames UDP. Explique para qué se utilizan. Qué es el pseudoheader?. Cómo se controlan los errores en los bits?

Puerto origen

Este campo es de utilidad cuando se requiere enviar una respuesta desde el receptor.

Puerto destino

Puerto mediante el cual se realiza la comunicación con la aplicación correspondiente.

Longitud UDP

Incluye el encabezado de 8 bytes y los datos.

Suma de verificación

Campo opcional. En caso de que no se calcule se almacena como 0.

El pseudoheader permite realizar una correcta suma de comprobación trayendo datos de la capa IP, como son, IP origen y destino, numero de protocolo y longitud de los datos. Esta información brinda protección antes paquetes perdidos.

bits	0 – 7	8 – 15	16 – 23	24 – 31
0	Dirección Origen			
32	Dirección Destino			
64	Ceros	Protocolo	Longitud UDP	
96	Puerto Origen		Puerto Destino	
128	Longitud del Mensaje		Suma de verificación	
160	Datos			

Para calcular el checksum, se suman todas las palabras de 16bits que forman el paquete y se almacena el resultado en el campo “suma de verificación”. Una vez que el mensaje es recibido se realiza este mismo calculo. Si la suma final son todos 1’s el mensaje es correcto. En caso contrario, el mensaje se rechaza.

Explique por qué UDP se adapta a ser utilizado en transmisiones multicast y en transmisiones multimedia (telefonía IP, videoconferencias, etc.). Comente las consecuencias que la aplicación masiva de UDP en ese tipo de aplicaciones podría tener para la red.

Se adapta a las transmisiones multicast porque es un protocolo no orientado a la conexión, de no ser así para realizar ese tipo de transmisiones se deberían inicializar tantas conexiones como destinatarios tenga el paquete y eso no sería viable.

Se adapta a transmisiones multimedia, pues al no utilizar retransmisión es su funcionamiento logra mayor fluidez a la hora del tiempo real. De hecho existen dos protocolos de tiempo real que lo utilizan como el rtp y el rtcp.

Si se hiciera masiva su aplicación, podría degradar el rendimiento en general de las aplicaciones, debido al retardo de los paquetes o a q los paquetes no lleguen a tiempo por la carga de la red.

Describa en qué consisten las siguientes características que presenta TCP: envío de datos urgentes, envío inmediato, ventanas deslizantes de longitud variable.

Envío de datos urgentes

Cuando un usuario interactivo oprime las teclas Supr o Ctrl+C para interrumpir una operación remota que ha iniciado, la aplicación emisora coloca información de control en el flujo de datos y se la da a TCP junto con el indicador URGENT. Este evento ocasiona que TCP interrumpa el encolamiento de datos y transmita inmediatamente todo lo que tenga para esa conexión.

Cuando el destino recibe los datos urgentes, se interrumpe la aplicación receptora (por ejemplo, se le da una señal en términos de UNIX), a fin de que pueda detener lo que esté haciendo y que el flujo de datos para

encontrar los datos urgentes. El final de los datos urgentes se marca para que la aplicación sepa cuándo terminan.

Envío inmediato

Algunas veces, la aplicación realmente necesita que los datos se envíen de inmediato a la máquina remota inmediatamente y que no se almacene en el búfer hasta que llegue la siguiente línea. Para obtener los datos, las aplicaciones pueden utilizar el indicador PUSH, que es una señal para TCP de que no debe retrasar la transmisión.

El bit PSH indica datos que se deben transmitir de inmediato. Por este medio se solicita atentamente al receptor que entregue los datos a la aplicación a su llegada y no los almacene en búfer hasta la recepción de un búfer completo (lo que podría hacer en otras circunstancias por razones de eficiencia).

Ventanas deslizante de longitud variable

Ver en próximas preguntas

Describa detalladamente el proceso de conexión TCP. Mencione qué campos del frame TCP se utilizan y cómo.

Para establecer una conexión, un lado, digamos el servidor, espera pasivamente una conexión entrante ejecutando las primitivas LISTEN y ACCEPT y especificando cierto origen o bien nadie en particular. El otro lado, digamos el cliente, ejecuta una primitiva CONNECT especificando la dirección y el puerto IP con el que se desea conectar, el tamaño máximo de segmento TCP que está dispuesto a aceptar y opcionalmente algunos datos de usuario (por ejemplo, una contraseña). La primitiva CONNECT envía un segmento TCP con el bit SYN encendido y el bit ACK apagado, y espera una

Respuesta. Al llegar el segmento al destino, la entidad TCP ahí revisa si hay un proceso que haya ejecutado un LISTEN en el puerto indicado en el campo de Puerto de destino. Si no lo hay, envía una respuesta con el bit RST encendido para rechazar la conexión. Si algún proceso está escuchando en el puerto, ese proceso recibe el segmento TCP entrante y puede entonces aceptar o rechazar la conexión; si la acepta, se devuelve un segmento de confirmación de recepción.

Describa el funcionamiento de la técnica “threewayhandshake” y explique con ejemplos cómo se evita crear dos conexiones cuando ambas partes realizan simultáneamente el requerimiento de conexión.

Mirar respuesta anterior.

En el caso en que dos hosts intentan simultáneamente establecer una conexión entre los mismos dos sockets, es que sólo se establece una conexión, no dos, pues las conexiones se identifican por sus puntos terminales. Si el primer establecimiento resulta en una conexión identificada por (x, y), al igual que en el segundo, sólo se hace una entrada de tabla, es decir, de (x, y).

Describa detalladamente el mecanismo de ventana deslizante utilizado por TCP. Mencione qué campos del frame TCP se utilizan y cómo.

El protocolo básico usado por las entidades TCP es el protocolo de ventana corrediza. Cuando un transmisor envía un segmento, también inicia un temporizador. Cuando llega el segmento al destino, la entidad TCP receptora devuelve un segmento (con datos, si existen, de otro modo sin ellos) que contiene un número de confirmación de recepción igual al siguiente número de secuencia que espera recibir. Si el temporizador del emisor expira antes de la recepción de la confirmación, el emisor envía de nuevo el segmento. El control de flujo en el TCP se maneja usando una ventana corrediza de tamaño variable.

El campo Tamaño de ventana del segmento indica la cantidad de bytes que pueden enviarse comenzando por el byte cuya recepción se ha confirmado. Es válido un campo de Tamaño de ventana de 0, e indica que se han recibido los bytes hasta Número de confirmación de recepción −1, inclusive, pero que el receptor actualmente necesita un descanso y quisiera no recibir más datos por el momento, gracias. El permiso para enviar puede otorgarse después enviando un segmento con el mismo Número de confirmación de recepción y un campo Tamaño de ventana distinto de cero. En los protocolos del capítulo 3, las confirmaciones de recepción de las tramas recibidas y los permisos para enviar nuevas tramas estaban enlazados. Ésta fue una consecuencia de un tamaño de ventana fijo para cada protocolo. En TCP, las confirmaciones de recepción y los permisos para enviar datos adicionales son completamente independientes. En efecto, un receptor puede decir: “He recibido bytes hasta k, pero por ahora no deseo más”. Esta independencia (de hecho, una ventana de tamaño variable) da flexibilidad adicional.

Describa detalladamente el proceso de desconexión TCP. Mencione qué campos del frame TCP se utilizan y cómo.

Cada conexión simplex se libera independientemente de su igual. Para liberar una conexión, cualquiera de las partes puede enviar un segmento TCP con el bit FIN establecido, lo que significa que no tiene más datos por transmitir. Al confirmarse la recepción del FIN, ese sentido se apaga. Sin embargo, puede continuar un flujo de datos indefinido en el otro sentido. Cuando ambos sentidos se han apagado, se libera la conexión. Si no llega una respuesta a un FIN en un máximo de dos tiempos de vida de paquete, el emisor del FIN libera la conexión.

Describa detalladamente el funcionamiento de la opción “Windows Scale” de TCP. Ejemplifique.

Se propuso una opción de escala de ventana que permite al emisor y al receptor negociar un factor de escala de ventana. Este número da la posibilidad de que ambos lados desplacen el campo de Tamaño de ventana hasta 14 bits a la izquierda, permitiendo por tanto ventanas de hasta 230 bytes. La mayoría de las implementaciones actuales de TCP manejan esta opción.

Describa detalladamente el funcionamiento de la opción “Timestamp” de TCP. Ejemplifique. Mencione para qué funciones se la puede utilizar.

Referente al tiempo medio de envío que un circuito experimenta. El tiempo medio de envío determinará de forma precisa cuanto esperará TCP antes de intentar retransmitir un segmento que no ha sido acusado de recibo.

Puesto que cada red tiene características de latencia únicas, TCP tiene que adaptarse a dichas características con idea de establecer umbrales de tiempo precisos para los temporizadores de acuse de recibo. Las redes de área local (LAN) sufren normalmente tiempos de latencia muy bajos, y así TCP podrá usar valores bajos para los temporizadores de acuse de recibo. Si un segmento no es acusado de recibo rápidamente, un emisor puede retransmitir los datos en cuestión rápidamente, minimizando así el ancho de banda perdido y el retraso.

Por el contrario, usar un umbral bajo en una red de área extensa (WAN) causará problemas simplemente porque los temporizadores de acuse de recibo expirarán antes de que los datos alcancen su destino.

De esta forma, para que TCP pueda establecer con exactitud el umbral de los temporizadores para un circuito virtual, tiene que medir el tiempo medio de envío para varios segmentos. Finalmente, tiene que monitorizar otros segmentos a lo largo de la duración de la conexión para mantenerse al día de los cambios producidos en la red. Y ahora es cuando la opción Timestamp aparece en escena.

De manera similar a la mayoría de las otras Opciones TCP tratadas aquí, la opción Timestamp debe ser enviada durante el 3-way handshake para habilitar su uso por los segmentos subsiguientes.

El Timestamp se compone de los campos Eco Timestamp (Timestamp Echo) y Respuesta Timestamp (Timestamp Reply), de los cuales el campo de respuesta es siempre puesto a cero por el emisor y completado por el receptor, después de lo cual es enviado de vuelta al emisor original. Ambos campos del Timestamp tiene 4 bytes de largo.

Explique la función de los bits PUSH, ACK, RST, SYN y URG del frame TCP

PUSH: las aplicaciones pueden utilizar el indicador PUSH, que es una señal para TCP de que no debe retrasar la transmisión.

ACK: El bit ACK se establece en 1 para indicar que el Número de confirmación de recepción es válido. Si el ACK es 0, el segmento no contiene una confirmación de recepción, por lo que se ignora el campo de Número de confirmación de recepción.

RST: El bit RST se usa para restablecer una conexión que se ha confundido debido a una caída de host u otra razón; también sirve para rechazar un segmento no válido o un intento de abrir una conexión.

SYN: se usa para denotar CONNECTION REQUEST y CONNECTION ACCEPTED, y el bit ACK sirve para distinguir entre ambas posibilidades.

URG(URGENT): Este evento ocasiona que TCP interrumpa el encolamiento de datos y transmita inmediatamente todo lo que tenga para esa conexión.

Explique la función de los bits ECE, CWR y NS

La extensión a los protocolos IP y TCP ECN(Explicit Congestion Notification) hace uso

de estos bits.

NS(Nonce Sum): is used to protect against accidental or malicious concealment of marked packets from the TCP sender.

ECE (ECN-Echo): Upon receiving an IP packet with the Congestion Experienced codepoint, the TCP receiver echoes back this Congestion indication using the ECE flag in the TCP header. When an endpoint receives a TCP segment with the ECE bit it reduces its Congestion window as for a packet drop.

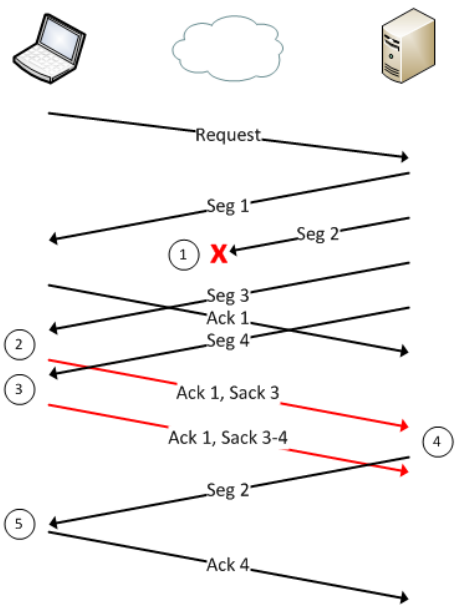
CWR (Congestion Windows Reduce): It then acknowledges the Congestion indication by sending a segment with the CWR bit set.

Describa el problema de los números de secuencia duplicados y cómo es resuelto en el contexto de TCP/IP.

Incompleta

Explique en detalle el funcionamiento de la opción SACK (selectiveacknowledgements) de TCP. De un ejemplo de la ganancia en performance obtenida al usarla.

Supongamos que de una secuencia de 4 paquetes, el cliente recibe el #1, #3, y el #4. El #2 no llego por alguna falla. En caso normal, el servidor recibe desde el cliente que el paquete 2 no llego correctamente. Por lo que el servidor, va a reenviar los paquetes 2,3 y 4. Con la opción SACK, el cliente puede indicar al servidor que recibió correctamente todos los paquetes salvo el 2.



La mejora de performance se produce ya que el reenvió de paquetes es mucho menor que de la manera convencional.

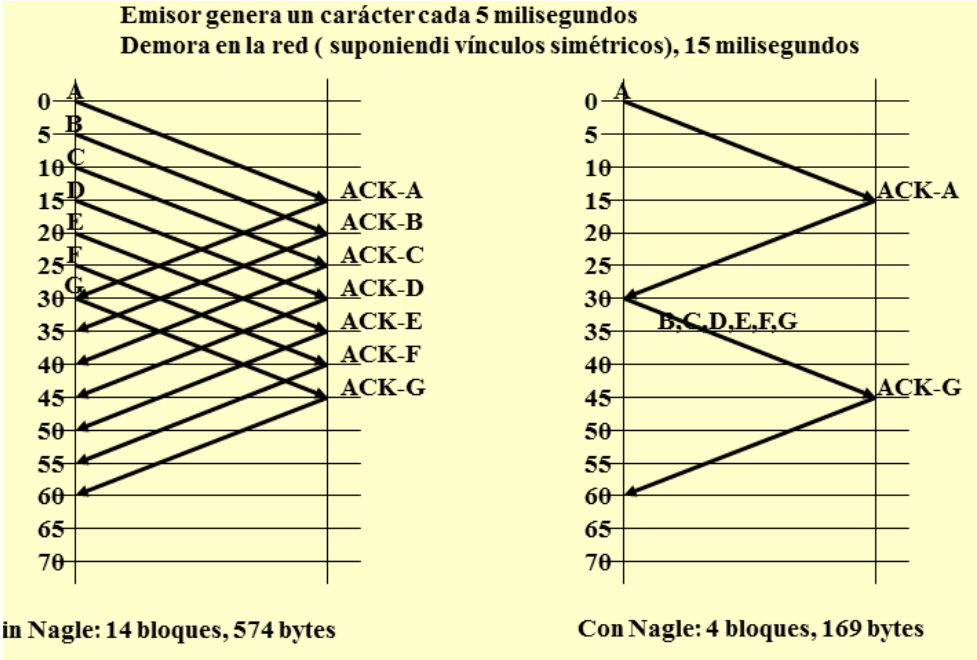
Explique en qué consiste y describa detalladamente (y con ejemplo) el algoritmo de Nagle.

Lo que sugirió Nagle es sencillo: al llegar datos al emisor un byte a la vez, simplemente se envía el primer byte y se almacena en búfer los demás hasta la confirmación de recepción del byte pendiente. Luego se transmiten todos los caracteres del búfer en un segmento TCP y nuevamente se comienzan a almacenar en búfer los datos hasta que se haya confirmado la recepción de todos. Si el usuario escribe con rapidez y la red es lenta, puede entrar una cantidad importante de caracteres en cada segmento, reduciendo en gran medida el ancho de banda usado. Además, el algoritmo permite el envío de un nuevo paquete si han entrado suficientes datos para llenar la mitad de la ventana o la totalidad de un segmento.

El algoritmo de Nagle se usa ampliamente en las implementaciones de TCP, pero hay veces en

que es mejor inhabilitarlo. En particular, al operar una aplicación X-Windows a través de Internet,

los movimientos del ratón tienen que enviarse a la computadora remota. (X-Windows es el sistema de ventanas que se utiliza en la mayoría de los sistemas UNIX.) Su acumulación para enviarlos en ráfagas hace que el movimiento del cursor sea errático, lo que no complace mucho a los usuarios.



Explique en que consiste y describa detalladamente (y con ejemplo) el problema conocido como “SillyWindowSyndrome”

Este problema ocurre cuando se pasan datos a la entidad emisora en bloques grandes, pero una aplicación interactiva del lado receptor lee datos a razón de 1 byte a la vez. Para ver el problema, observe la figura 6-35. Inicialmente, el búfer TCP del lado receptor está lleno y el emisor lo sabe (es decir, tiene un tamaño de ventana de 0). Entonces la aplicación interactiva lee un carácter del flujo TCP. Esta acción hace feliz al TCP receptor, por lo que envía una actualización de ventana al emisor indicando que está bien que envíe 1 byte. El emisor accede y envía 1 byte. El búfer ahora está lleno, por lo que el receptor confirma la recepción del

segmento de 1 byte pero establece la ventana en 0. Este comportamiento puede continuar indefinidamente. La solución de Clark es evitar que el receptor envíe una actualización de ventana para 1 byte.

En cambio, se le obliga a esperar hasta tener disponible una cantidad de espacio, y luego lo anuncia. Específicamente, el receptor no debe enviar una actualización de ventana hasta que pueda manejar el tamaño máximo de segmento que anunció al establecerse la conexión, o que su búfer quede a la mitad de capacidad, lo que sea más pequeño.

Además, el emisor también puede ayudar al no enviar segmentos muy pequeños. En cambio, debe intentar esperar hasta haber acumulado suficiente espacio en la ventana para enviar un segmento completo, o cuando menos uno que contenga la mitad del tamaño de búfer del receptor (que debe estimar a partir del patrón de las actualizaciones de ventana que ha recibido anteriormente).

Describa en general como funciona y qué tiene en cuenta el control de congestión realizado por TCP. Incluya los aspectos relativos al receptor y a la red. (Causas, detección y prevención de congestión)

Cuando la carga ofrecida a cualquier red es mayor que la que puede manejar, se genera una congestión. Además debido a la interacción con otros elementos se produce una reacción en cadena que hace que el problema sea más grave. Cuando el problema se hace demasiado grande se produce la caída de la red o deadlock. Otra posible causa es la saturación del receptor lo que produce un desborde y la perdida de paquetes.

La idea es no inyectar un paquete nuevo en la red hasta que salga uno viejo (es decir, se entregue). El TCP intenta lograr esta meta manipulando dinámicamente el tamaño de la ventana.

Todos los algoritmos TCP de Internet suponen que las expiraciones de tiempo son causadas por congestión y las revisan en busca de problemas. Otra forma de detección es el exceso de ack duplicados.

Para evitar que ocurra. Al establecerse una conexión, se tiene que seleccionar un tamaño de ventana adecuado. El receptor puede especificar una ventana con base en su tamaño de búfer. Si el emisor se ajusta a su tamaño de ventana, no ocurrirán problemas por desbordamiento de búferes en la terminal receptora, pero aún pueden ocurrir debido a congestión interna en la red.

La solución de Internet es aceptar que existen dos problemas potenciales (capacidad de la red y capacidad del receptor) y manejarlos por separado. Para ello, cada emisor mantiene dos ventanas: la ventana que ha otorgado el receptor y una segunda ventana, la ventana de congestión. Cada una refleja la cantidad de bytes que puede enviar el emisor. La cantidad de bytes que pueden enviarse es la cifra menor de las dos ventanas. Por tanto, la ventana efectiva es el mínimo de lo que el emisor piensa que es correcto y lo que el receptor piensa que está bien. Si el receptor dice “envía 8 KB” pero el emisor sabe que las ráfagas de más de 4 KB saturan la red, envía 4 KB. Por otra parte, si el receptor dice “envía 8 KB” y el emisor sabe que las ráfagas de hasta 32 KB pueden llegar sin problemas, envía los 8 KB solicitados.

Describa que son, para qué se utilizan y cómo funcionan los mecanismos de control de congestión “slowstart” y “congestion avoidance”

Slowstart

Al establecer una conexión, el emisor asigna a la ventana de congestión el tamaño de segmento máximo usado por la conexión; entonces envía un segmento máximo. Si se recibe la confirmación de recepción de este segmento antes de que expire el temporizador, el emisor agrega el equivalente en bytes de un segmento a la ventana de congestión para hacerla de dos segmentos de tamaño máximo, y envía dos segmentos. A medida que se confirma cada uno de estos segmentos, se aumenta el tamaño de la ventana de congestión en un segmento máximo. Cuando la ventana de congestión es de n segmentos, si de todos los n se reciben confirmaciones de recepción a tiempo, se aumenta el tamaño de la ventana de congestión en la cuenta de bytes correspondiente a n segmentos. De hecho, cada ráfaga confirmada duplica la ventana de congestionamiento.

La ventana de congestión sigue creciendo exponencialmente hasta ocurrir una expiración del temporizador o alcanzar el tamaño de la ventana receptora. La idea es que, si las ráfagas de 1024, 2048 y 4096 bytes funcionan bien, pero una ráfaga de 8192 produce una expiración del temporizador, la ventana de congestión debe establecerse en 4096 para evitar la congestión. Mientras el tamaño de la ventana de congestión permanezca en 4096, no se enviará una ráfaga de mayor longitud, sin importar la cantidad de espacio de ventana otorgada por el receptor. Este algoritmo se llama arranque lento, pero no es lento en lo absoluto (Jacobson, 1988); es exponencial, y se requiere que todas las implementaciones de TCP lo manejen.

Congestion avoidance

Suposición de partida: sólo el 1% de los datagramas que se pierden en Internet (es decir, que deben ser retransmitidos) lo son por sufrir alteraciones en sus contenidos. La inmensa mayoría de las pérdidas son

provocadas por la congestión de determinadas redes.

– Principios de actuación:

- Cuando comienza a transmitir datos, un módulo TCP inicializa cwnd a 1 y sigue los dictados de Slow-Start. El ritmo crece exponencialmente por el incremento de cwnd en 1 por ack llegado (zona de Slow-Start - SS).
- TCP sigue transmitiendo según las pautas de SS hasta que detecta congestión (se produce la primera retransmisión por time-out).
- Entonces se baja a nuevamente la cwnd a 1 y se vuelve a actuar según SS (incremento de cwnd en una unidad por ack recogido)
- Cuando el ritmo de transmisión llega a un cierto umbral, se cambia a un crecimiento lineal del mismo, para evitar enviar demasiados datagramas a las redes (zona de Congestion Avoidance -CA) y congestionarlas.
- Al mismo tiempo, si en algún momento el transmisor retransmite un segmento por time-out, se reinicializa el valor de la ventana cwnd a 1 segmento y comienza a transmitirse según SS partiendo de la situación inicial. Se hace así para disminuir drásticamente el ritmo en cuanto se detecta congestión.

Describa detalladamente de qué manera combina TCP los métodos “slowstart” y “congestionavoidance”. Mencione variables involucradas, tiempos, etc.

• Fundamentos:

– Se gestionan tres variables (todas en num. de bytes):

- cwnd: ventana de congestión.
- ssthresh: umbral de tamaño slow-start. Fija umbral a partir del cual se deja SS y se pasa a CA.
- awnd: ventana anunciada por receptor.

– Algorítmo combinado control de flujo y de congestión:

- Inicio: cwnd = MSS (1 segmento) ; ssthresh = 65535 bytes
- Transmisor NUNCA envía más segmentos que el mínimo de cwnd y la ventana anunciada por el receptor. A ese mínimo le llamaremos ventana de transmisión actual (vta = min(cwnd,awnd))

• SI hay congestión (time-out de datos transmitidos O ack duplicados)

– ssthresh = max(1/2 vta,2xMSS).Mitad de ventana de transmisión actual (2 segmentos como mínimo).

– Y si la congestión se detecta por un time-out, cwnd=1

• SI llega un ACK

– SI cwnd <= ssthresh (se está en fase Slow Start -SS) cwnd=cwnd+MSS

– SINO (se está en fase Congestion Avoidance -CA) cwnd = cwnd + (MSSxMSS)/cwnd

- NOTA: implementaciones de algunas versiones de BSD en la fase CA hacen cwnd = cwnd + (MSSxMSS)/cwnd + MSS/8

Otra fuente

Se ha de tener en consideración el valor del slow start threshold size (sstresh), que no es más que una variable que se encarga de indicar el tamaño máximo de la ventana de transmisión, que se inicializa en este algoritmo a 65535 bytes.

Cada vez que se detecte congestión se actualizará el valor de este campo a la mitad del mínimo entre window ("win") y Congestion window ("cwnd").

Si el motivo de la congestión es debido a la expiración del tiempo para la confirmación de la recepción de un segmento (vence el temporizador de retransmisión), se pondrá el valor de ("cwnd") a MSS (Maximum segment size), lo que equivale a resetear la ventana de transmisión de comienzo lento.

Cuando se incrementa ("cwnd") siempre se tiene en cuenta el valor de ("sstresh"), de modo que si el valor de ("cwnd") no ha superado al de ("sstresh"), la forma de incrementarse será del mismo modo que se hace en

comienzo lento que no es otra forma que exponencial, mientras que si se ha superado el incremento será lineal.

cwnd(superado sstresh)=cwnd(anterior)+MSS/cwnd(antiguo)

En qué consisten los métodos “fast retransmit” y “fast recovery” utilizados por TCP?. Describa en detalle cómo se los utiliza y qué beneficios se obtienen de su uso.

Para remediar el problema de los largos períodos de inactividad atados al timeout, se ha introducido el mecanismo de Fast Retransmit. Con este mecanismo el TCP no tiene innecesariamente que esperar a que venza el timeout para poder retransmitir un segmento. Está previsto que el receptor mande el ACK relativo a un segmento, aunque no se hayan recibido aún los segmentos anteriores a éste. El transmisor deduce del número de ACK duplicados qué recibe, si el segmento se ha perdido o bien está sufriendo solamente retrasos. En efecto, un ACK duplicado indica el hecho tal que el receptor no puede mandar un ACK relativo a un segmento llegado fuera de orden, ya que está esperando recibir de uno precedente.

El mecanismo de Fast Retransmit prevé, que en caso de se reciban que 3 ACK duplicados (3DUPACK) relativos al mismo segmento, se procede a la retransmisión del segmento, sin esperar a que venza el timeout. Este mecanismo permite beneficios en el throughput y reduce el número de timeout.

La versión del TCP denominada TCP Reno, después de haber recibido 3DUPACK, pone el SlowStartThreshold a la mitad del valor actual de la ventana de congestión, pero en vez de reducir bruscamente este última hasta llevarla al valor de 1 segmento, la reduce a la mitad solamente, eliminando así la fase de Slow Start siguiente ("retomada veloz", fast recovery).

El motivo que lleva al control de congestión del TCP Reno a comportarse de forma distinta después de un acontecimiento de timeout y después de la recepción de tres ACK duplicados es que, en el segundo caso, aunque se haya perdido un paquete, la llegada de los 3DUPACK al remitente testimonia que la red es capaz de entregar, al menos, algún segmento, aunque otros se hayan perdido a causa de la congestión. La recepción de los tres ACK duplicados por lo tanto, representa en todo caso un síntoma del alto nivel de congestión de la red, pero menos grave que los timeout.

Mencione y describa brevemente los timers utilizados por TCP (retransmission, persist, keepalive y 2MSL).

Temporizador de retransmisión. Al enviarse un segmento, se inicia un temporizador de retransmisiones. Si la confirmación de recepción del segmento llega antes de expirar el temporizador, éste se detiene. Si, por otra parte, el temporizador termina antes de llegar la confirmación de recepción, se retransmite el segmento (y se inicia nuevamente el temporizador).

Persist timer: diseñado para evitar el siguiente bloqueo irreversible. El receptor envía una confirmación de recepción con un tamaño de ventana de 0, indicando al emisor que espere. Después, el receptor actualiza la ventana, pero se pierde al paquete con la actualización. Ahora, tanto el emisor como el receptor están esperando que el otro haga algo. Cuando termina el temporizador de persistencia, el emisor envía un sondeo

al receptor. La respuesta al sondeo da el tamaño de la ventana. Si aún es de cero, se inicia el temporizador de persistencia nuevamente y se repite el ciclo. Si es diferente de cero, pueden enviarse datos.

Keepalive timer: Cuando una conexión ha estado inactiva durante demasiado tiempo, el temporizador de seguir con vida puede expirar, haciendo que un lado compruebe que el otro aún está ahí. Si no se recibe respuesta, se termina la conexión. Esta característica es motivo de controversias porque agrega sobrecarga y puede terminar una conexión saludable debido a una partición temporal de la red.

2MSL: TCP después de realizar un "active close" y enviar el ACK del FIN debe esperar un tiempo TIME_WAIT. Porque permite a TCP reenviar el ACK del FIN si éste se pierde (el otro extremo, en este caso, reenviará el FIN cuando venza su temporizador). Hasta que no finalice este temporizador no se liberan el par de sockets de la conexión. Cada implementación selecciona un valor de MSL (Maximun Segment Lifetime): típicamente 2 minutos o 1 minuto o 30 segundos.

Describa detalladamente en qué consiste y la utilidad del timer "Keepalive Timer" de TCP.

Anteriormente explicado.

Describa detalladamente en qué consiste y la utilidad del timer "Persist Timer" de TCP.

Anteriormente explicado.

Describa detalladamente en qué consiste y la utilidad del timer "2MSL" de TCP.

Anteriormente explicado.

Describa detalladamente cómo se mide el tiempo de ida y vuelta (RTT) y qué consideraciones deben ser tenidas en cuenta.

La solución al problema de la elección de un buen temporizador de reenvío es usar un algoritmo muy dinámico que ajuste constantemente el intervalo de expiración del temporizador, con base en mediciones continuas del desempeño de la red. El algoritmo que generalmente usa el TCP lo debemos a Jacobson (1988) y funciona como sigue. Por cada conexión, el TCP mantiene una variable, RTT(round-trip time), que es la mejor estimación actual del tiempo de ida y vuelta al destino en cuestión. Al enviarse un segmento, se inicia un temporizador, tanto para ver el tiempo que tarda la confirmación de recepción como para habilitar una retransmisión si se tarda demasiado. Si llega la confirmación de recepción antes de expirar el temporizador, el TCP mide el tiempo que tardó la confirmación de recepción, digamos M. Entonces actualiza RTT de acuerdo con la fórmula.

RTT $=\alpha RTT+(1-\alpha)M$ donde α es un factor de amortiguamiento que determina el peso que se le da al valor anterior. Por lo común, $\alpha = 7/8$.

Describa en detalle de qué manera se estima el tiempo de retransmisión (RTO) en TCP a partir del RTT. Explique las consideraciones que se toman en cuenta para lograr una estimación correcta.

Aun dado un buen valor de RTT, la selección de una expiración adecuada del temporizador de retransmisión no es un asunto sencillo. Normalmente el TCP usa βRTT, pero el truco es seleccionar β. En las implementaciones iniciales, β siempre era 2, pero la experiencia demostró que un valor constante era inflexible puesto que no respondía cuando subía la variación.

En 1988, Jacobson propuso hacer queβ fuera aproximadamente proporcional a la desviación estándar de la función de densidad de probabilidad del tiempo de llegada de las confirmaciones de recepción, por lo que una variación grande significa unaβ grande, y viceversa. En particular, sugirió el uso de la desviación media como una forma rápida de estimar la desviación estándar. Su algoritmo requiere mantener otra variable amortiguada, D, la desviación. Al llegar una confirmación de recepción, se calcula la diferencia entre el valor esperado y el observado, RTT− M. Un valor amortiguado de esta cifra se mantiene en D mediante la fórmula

D= αD+(1 − α) RTT−M donde α puede ser o no el mismo valor usado para amortiguar RTT. Si bien D no es exactamente igual a la desviación estándar, es bastante buena, y Jacobson demostró la manera de calcularla usando sólo sumas, restas y desplazamientos de enteros, lo que es una gran ventaja. La mayoría de las implementaciones TCP usan ahora este algoritmo y establecen el intervalo de expiración del temporizador en

Expiración del temporizador =RTT+4× D

La selección del factor 4 es un tanto arbitraria, pero tiene dos ventajas. Primera, puede hacerse la multiplicación por 4 con un solo desplazamiento. Segunda, reduce al mínimo las expiraciones de temporizador y las retransmisiones innecesarias porque menos del 1% de todos los paquetes llegan más de cuatro desviaciones estándar tarde.

Explique qué es un socket, cuál es su utilidad y cómo se lo utiliza.

Un socket es un mecanismo de entrega de paquetes entre dos procesos o hilos dentro de una red. Un socket se define como el punto final en una comunicación entre cliente y servidor. Un socket queda definido por una dirección IP y un número de puerto determinado.

Los sockets permiten implementar una arquitectura cliente servidor, indicando mediante la definición antes vista, cual es la maquina con la cual debe comunicarse y a través de que puerto. De esta manera, se permite tanto al cliente como al servidor leer y escribir la información.

Primitiva	Significado
SOCKET	Crea un nuevo punto terminal de comunicación
BIND	Adjunta una dirección local a un *socket*
LISTEN	Anuncia la disposición a aceptar conexiones; indica el tamaño de cola
ACCEPT	Bloquea al invocador hasta la llegada de un intento de conexión
CONNECT	Intenta establecer activamente una conexión
SEND	Envía datos a través de la conexión
RECEIVE	Recibe datos de la conexión
CLOSE	Libera la conexión

La primitiva SOCKET crea un nuevo punto de comunicación y le asigna espacio en las tablas de la entidad de transporte. Los parámetros de la llamada especifican el formato de direccionamiento que se utilizará, el tipo de servicio deseado (por ejemplo, flujo confiable de bytes) y el protocolo.

Los sockets recién creados no tienen direcciones de red. Éstas se asignan mediante la primitiva BIND. Una vez que un servidor ha destinado una dirección a un socket, los clientes remotos pueden conectarse a él.

A continuación viene la llamada LISTEN, que asigna espacio para poner en cola las llamadas entrantes por si varios clientes intentan conectarse al mismo tiempo.

Para bloquearse en espera de una conexión entrante, el servidor ejecuta una primitiva

ACCEPT. Cuando llega una TPDU solicitando una conexión, la entidad de transporte crea un socket nuevo con las mismas propiedades que el original y devuelve un descriptor de archivo para él. ACCEPT regresa un descriptor de archivo normal, que puede utilizarse para leer y escribir de la forma estándar, al igual que con los archivos.

Del lado del cliente también debe crearse un socket con la primitiva SOCKET, pero no se requiere BIND, ya que la dirección utilizada no le importa al server. La primitiva CONNECT bloquea al invocador y comienza activamente el proceso de conexión.

Así se establece la comunicación y ambos procesos pueden enviar y recibir información.

Describa los diferentes tipos de sockets (datagram, stream, etc.)

Datagram sockets: es un tipo de socket no orientado a la conexión, que funciona como punto de envío o recepción de paquetes. Cada uno de los paquetes que manejan estos sockets se direccionan y rutean de manera individual. Para recibir paquetes broadcast, un datagram socket debe tener especificada la dirección desde la cual leer la información.

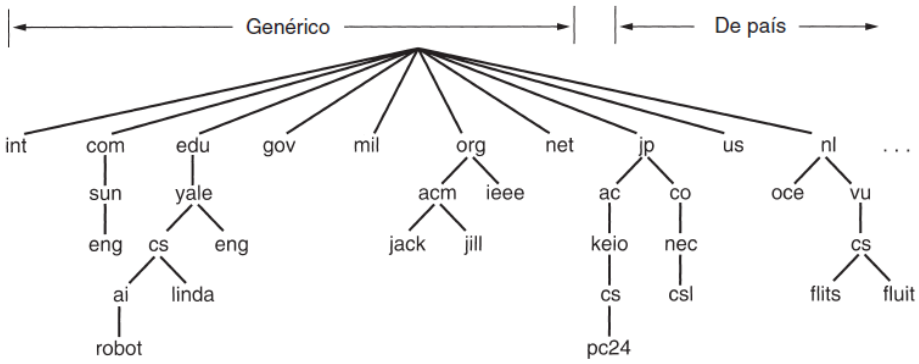
Stream sockets: es un tipo de socket orientado a la conexión que provee un flujo de datos secuencial y sin duplicados, con mecanismos bien definidos para la creación y destrucción de conexiones, como así también para la detección de errores.

Socket: ¿que es un dominio y qué es una familia de direcciones? Mencione dominios y familias de direcciones que conozca.

Un dominio es una red de identificación asociada a un grupo de dispositivos o equipos conectados a la red Internet. El propósito principal de los dominios, junto con DNS, es traducir direcciones IP a términos

memorizables y fáciles de recordar. Esto permite que cualquier servicio cambie de lugar dentro de la red, y el cambio sea transparente a los usuarios.

Conceptualmente, Internet se divide en 200 dominios de nivel superior, cada uno de los cuales abarca muchos hosts. Cada dominio se divide en subdominios, los cuales, a su vez, también se dividen, y así sucesivamente. Todos estos dominios pueden representarse mediante un árbol, como se muestra en la figura.



Las hojas del árbol representan los dominios que no tienen subdominios (pero que, por supuesto, contienen máquinas). Un dominio de 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 y de país. Los dominios genéricos originales son com (comercial), edu (instituciones educativas), gov (el gobierno federal de Estados Unidos), int (ciertas organizaciones internacionales), mil (las fuerzas armadas de Estados Unidos), net (proveedores de red) y org (organizaciones no lucrativas). Los dominios de país incluyen una entrada para cada país.

¿En qué consiste la parametrización de sockets?. Mencione 5 parámetros especificando su función.

La parametrización de sockets sirve para indicar cuales son los participantes de la comunicación de datos. Dentro de los parámetros posibles para un sockets están:

1. Un puerto de protocolo local que específica donde recibe mensajes un proceso.

2. La dirección del host local, la cual identifica al anfitrión que recibirá los paquetes de datos.

3. Un puerto de protocolo remoto que identifica al proceso destino.

4. La dirección del anfitrión remoto, que identifica al anfitrión destino.

5. Un protocolo, que define como los procesos transfieren la información a través de la red.

Ejemplo de 5 funciones que utilicen estos parámetros especificados:

- Creación de un socket:

  socket(protocol_familiy, socket_type, protocol)

- Conexión de un socket:

Connect (socket_handle, remote_socket_address, address_length);

- Asignación de nombre:

  Bind (socket_handle, local_socket_address, address_lenght);