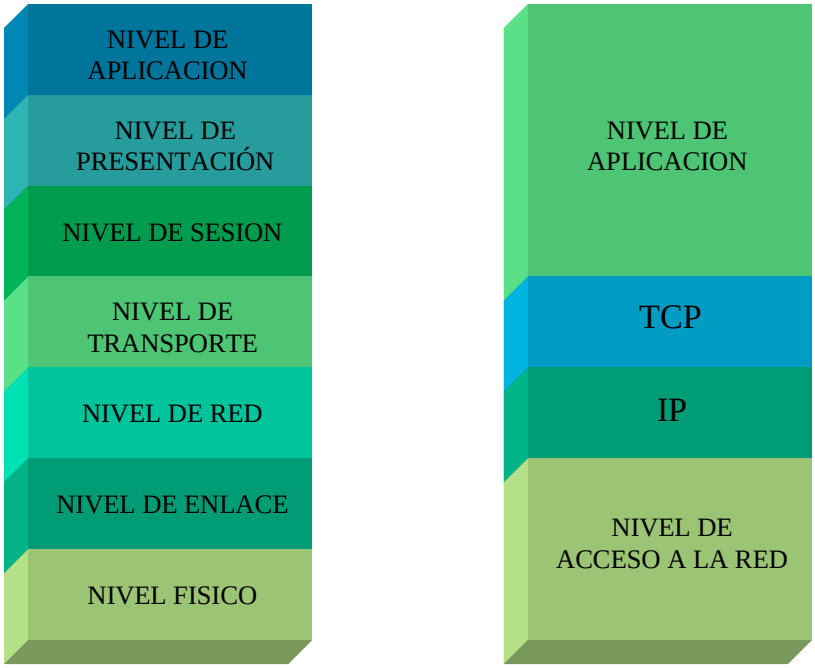
CAPÍTULO 1

Encaminamiento

1.1. Introducción

La capa de Red (modelo OSI) se encarga de llevar los paquetes desde el origen hasta el destino. Llegar al destino puede requerir muchos saltos por enrutadores intermedios. Dicha función contrasta con la que realiza la capa de enlace de datos, la cual se encarga de mover los marcos desde el emisor hasta el receptor. En consecuencia, la capa de red es la de nivel más bajo que maneja la transmisión de extremo a extremo.



Comparación entre los modelos de capas OSI y TCP/IP

Este nivel permite la transferencia de datos entre sistemas finales a través de uno o varios tipos de redes de datos. Así, los niveles superiores no necesitan saber nada sobre cómo se realiza la transmisión en los niveles inferiores ni de la tecnología de conmutación utilizada

Eliminado: . .

Eliminado: en Redes IP

Eliminado: 89¶

para conectar los sistemas. En este nivel, el sistema establece un diálogo con la red para especificar la dirección del destino y para solicitar ciertas facilidades de la red, como por ejemplo, la prioridad.

La función principal del nivel de red es la *encaminamiento*, que depende del tipo de red o redes que haya entre los sistemas finales y que incluso puede no existir si el medio de transmisión no la necesitase. Con esta función, el nivel de red ha de conseguir que la información llegue de la máquina origen a la de destino. En una red existe un gran número de entidades funcionando para que el encaminamiento se realice de una forma eficiente. También se debe resolver el problema de la interconexión de redes, dado que puede ser que utilicen distintos esquemas de direccionamiento, formatos de sus esquemas de direccionamiento, formatos de sus UDP's, etc. A la UDP del nivel de red se la llama paquete.

El nivel de red oculta a las entidades de transporte cómo se emplean los recursos inferiores para obtener conexiones de red. Así, el nivel de transporte se ocupa sólo de la calidad de servicio y su coste, no si se dispone de una red local, o de conmutación de paquetes o si la comunicación es vía satélite o por medio de una fibra óptica. El nivel de red proporciona las siguientes facilidades

Con formato

Con formato: Numeración y viñetas

- Direcciones de red
- Conexiones de red
- Notificación de errores
- Secuenciamiento
- Control de flujo
- Reinicio
- Liberación de servicios

Hay que hacer notar que algunas de estas facilidades son opcionales.

Dentro de las funciones del nivel de red, hay que destacar las siguientes:

- Encaminamiento y retransmisión

Con formato: Numeración y viñetas

- Conexiones de red
- Multiplexación de las conexiones de red
- Segmentación y bloqueo
- Detección de errores
- Recuperación de errores
- Secuenciamiento
- Control de flujo
- Reinicio
- Selección de servicio
- Gestión del nivel de red

Estas funciones son necesarias para que las entidades del nivel de red puedan ofrecer los servicios antes descritos al nivel de transporte.

Para ir recorriendo la ruta marcada por el nivel de red habrá que ir pasando de nodo a nodo a través de los enlaces “directos” que haya entre éstos. Los dos restantes niveles son los que se encargan del enlace entre el sistema y el medio de comunicación (la red de datos). En primer lugar, será necesario que la arquitectura permita el uso de una gran variedad de diferentes medios físicos con diferentes procedimientos de control, y es de esta manera como surge el nivel 2 o de enlace.

Para poder llevar los paquetes del origen hacia el destino, la capa de red debe conocer la topología de la subred de comunicación, es decir la conexión entre enrutadores y escoger las trayectorias adecuadas a través de ellos. Hay que también escoger la mejor ruta (enrutadores y líneas de comunicación) evitando sobrecargar una de ellas y que las otras estén desocupadas. Por otro lado cuando las redes a conectar tienen diferentes plataformas, hay que saber responder ante los problemas que causa dicho parámetro.

1.1.1. Variables de diseño en la Capa de red

Los principales problemas que se deben analizar el momento de diseñar la capa de red son: los servicios que se prestan a la capa de transporte y la estructura interna de la subred.

Eliminado: 89¶

1.1.1.1. Servicios proporcionados a la capa de transporte

La interfaz capa de red/capa de transporte es la encargada de dar los servicios proporcionados por la capa de red hacia la de transporte, además es de mucha importancia por ser la interfaz entre el cliente y la portadora, es decir el límite de la subred, razón por la cual debe ser muy bien definida.

Los objetivos que deben cumplir los servicios de la capa de red son:

- Los servicios deben ser independientes de la tecnología de subred.
- La capa de transporte debe estar aislada de la cantidad, tipo y topología de las subredes presentes.
- Las direcciones de red disponibles para la capa de transporte deben seguir un plan de numeración uniforme, aún a través de varias LAN y WAN

Cumplidos estos objetivos el inconveniente surge si la capa de red debe proporcionar servicios orientados a conexión o servicios no orientados a conexión. Aquellos que se inclinan por un servicio orientado a conexión son los de las compañías telefónicas y los que se inclinan por los no orientados a conexión es la comunidad Internet.

Servicios orientados a conexión

Ventajas:

- Confiabilidad en la entrega de la información
- Confirmación de la entrega
- No hay paquetes perdidos
- No hay paquetes duplicados
- Los paquetes adicionan solamente un número de circuito virtual
- Se establece una QoS

Eliminado: 89¶

Desventajas:

- La complejidad está en la capa de red
- Tiempos de conexión sin utilización
- Tarifación por todo el tiempo de establecimiento de la conexión
- Si se cae un enrutador todos los CV´s se pierden, y por tanto los paquetes

Servicios no orientados a conexión

Ventajas:

- La complejidad está en la capa de transporte (hosts controlan errores y flujo)
- La capa de red sólo se limita a entregar y recibir paquetes
- Cada paquetes se transporta independientemente de sus antecesores
- Tarifación solamente por el tiempo de intercambio de información
- La capacidad de cómputo de los hosts se ha vuelto barata
- La subred es una inversión (inter)nacional que durará décadas por lo que no hay que cargarla de características e implementaciones
- En aplicaciones como la voz digitalizada y video en tiempo real es preferible la entrega rápida que la entrega exacta
- Si se cae un enrutador, se pueden establecer rutas alternativas

Eliminado: o

Eliminado: ´

Desventajas:

- No confiabilidad en la entrega de la información
- Los paquetes pueden adicionar mucha información (direcciones de camino)
- Los paquetes pueden perderse y/o duplicarse

1.1.1.2. Estructura Interna de la capa de red

Existen dos filosofías, la una que utiliza conexiones y otra que funciona sin conexiones. La que utiliza conexiones establece una conexión llamada *circuito virtual* en analogía al sistema telefónico; y la que trabaja sin conexiones utiliza unos paquetes denominados *datagramas*, llamados así en analogía con los telegramas.

Subred con Circuitos Virtuales

Se utilizan en subredes con servicio orientado a conexión y se crean cuando se establece una conexión y dejan de existir cuando se termina la conexión. Utilizan una sola ruta predeterminada para la transmisión.

Los enrutadores deben poseer una tabla con una entrada y salida por circuito virtual abierto que pasa a través suyo. Cada paquete debe poseer en su cabecera el número de CV, el numero secuencial correspondiente, etc. Por tanto cada enrutador sabrá por qué salida deberá enviar dichos paquetes.

Subred con datagramas

En una red con datagramas, los paquetes pueden enviarse por diferentes rutas, inclusive si existe un servicio orientado a conexión. Aunque las subredes de datagramas necesitan realizar mayor trabajo, estas suelen ser más robustas que las de CV´s y son más adaptables a los inconvenientes que se pueden presentar.

Los enrutadores en este caso poseen una tabla que indica la línea de salida para comunicarse con cada enrutador adyacente que llevara al destino el paquete

Eliminado: 89¶

1.1.1.3. Circuitos Virtuales vs. Datagramas

Acción	Datagramas	Circuitos Virtuales
Establecimiento del circuito	No necesaria	Requerida
Direccionamiento	Cada paquete contiene completas la dirección de origen y de destino	Cada paquete contiene un número de Circuito Virtual corto
Información de estado	La subred no contiene información de estado	Cada CV requiere espacio de la tabla de subred
Enrutamiento	Cada paquete se enruta independientemente	Ruta escogida cuando se establece el CV; todos los paquetes siguen esa ruta
Efecto de fallas del enrutador	Ninguno, exceptuando paquetes perdidos durante una caída	Terminan todos los CV que pasan a través del enrutador con falla
Control de congestionamientos	Difícil	Fácil si pueden asignarse por adelantado buffers suficientes a cada CV

Comparación de las subredes de datagramas y de circuitos virtuales

Espacio de memoria del enrutador y ancho de banda

En las subredes de CV´s las cabeceras de los paquetes solo contienen un número de circuito virtual, mientras que en las de Datagramas contienen la dirección completa del origen y destino que podrían resultar excesiva carga y un ancho de banda desperdiciado. Por el contrario el costo por el uso de CV es el espacio en la tabla de los enrutadores. Mucho va a depender entonces el “costo” de las líneas de comunicación y de la memoria del enrutador.

Eliminado: cebeceras

Eliminado: b

Tiempo de establecimiento contra tiempo de análisis de la dirección

El establecimiento de un CV consume tiempo, sin embargo todo lo que hay que hacer para transportar los datos es fácil, utilizar el número de CV, buscar en una tabla indexada y encontrar el origen; en una subred de datagramas es más complicado.

Congestionamientos

En una subred de CV el control de congestionamiento es mucho más sencillo puesto que previamente se ha establecido un camino para el transporte de datos que tiene reservado el

ancho de banda y la capacidad de enrutamiento. En una subred de datagramas es complicado porque existen varias rutas para el transporte.

Sistemas de procesamiento de transacciones

Cuando se utiliza un POS para realizar transacciones con tarjetas de crédito, se tarda mucho más tiempo en el establecimiento de un CV que en realizar una transacción, siendo un tráfico de este tipo, las subredes con CV´s resultan ineficientes.

Vulnerabilidad

En una subred con CV´s cuando cae un enrutador, se pierde su memoria, por tanto se terminan todos los CV´s que pasan por el. Cuando es una subred con Datagramas solo se perderán los paquetes encolados en el enrutador, dependiendo si han sido reconocidos o no.

Cabe indicar que el servicio ofrecido, sea orientado a conexión o no orientado a conexión, es independiente de la estructura de la subred. Esto significa que las cuatro combinaciones posibles podrían darse en la realidad. El caso de una subred de Circuitos Virtuales sobre un servicio no orientado a conexión es el ejemplo de IP sobre ATM. En la siguiente tabla se muestran ejemplos de los cuatro casos

	Datagramas	**Circuitos Virtuales**
No orientada a conexión	UDP sobre IP	UDP sobre IP sobre ATM
Orientada a conexión	TCP sobre IP	ATM AAL1 sobre ATM

Ejemplos de combinaciones de servicio y estructuras de la capa de red

Eliminado: 89¶

CAPÍTULO 2

Algoritmos de Encaminamiento

Eliminado: 1.2.

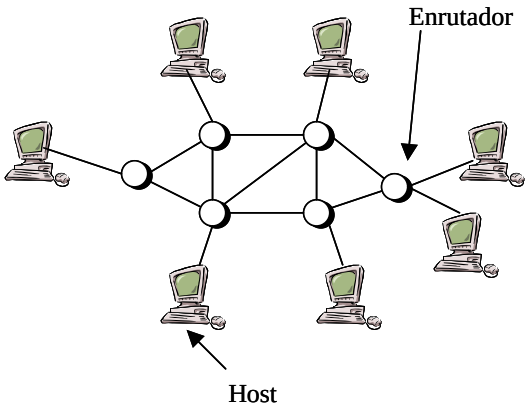
Eliminado: e

2.1. Introducción

Una de las principales funciones que cumple la capa de red es la de enrutar paquetes desde la máquina origen hasta la máquina destino, dicha ruta puede incluir varios saltos o enrutadores; inclusive en las redes broadcast es necesario encaminar los paquetes cuando el origen y el destino no están en la misma red. Los algoritmos que escogen las rutas y las estructuras de datos que éstos usan son un área principal del diseño de la capa de red.

El *algoritmo de encaminamiento o enrutamiento* es aquella parte del software de la capa de red encargada de decidir la línea de salida por la que se transmitirá un paquete de entrada. Si la subred usa datagramas internamente, esta decisión de enrutamiento debe hacerse cada vez que llega un paquete de datos de entrada, dado que la mejor ruta podía haber cambiado desde la última vez. Si la subred usa circuitos virtuales internamente, las decisiones de enrutamiento se toman solo al establecerse un circuito virtual nuevo.

Existen algunas propiedades que los algoritmos de encaminamiento deberían tener: corrección (calcular los caminos más óptimos), sencillez, robustez (funciona correctamente durante mucho tiempo y sin degradaciones), estabilidad (si cae un enlace busca otro camino), equitatividad (tratar de reducir a lo menos el tiempo de transmisión de la información y optimilidad (buscar el máximo de rendimiento). Hay que tener mucho cuidado con estas dos últimas características que podrían resultar contradictorias



Eliminado: infromaación

Eliminado: 89¶

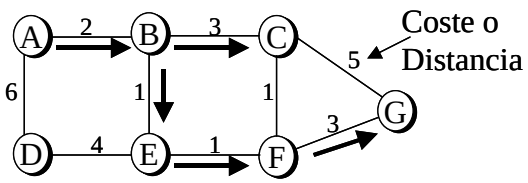
Eliminado: ¶

En consecuencia: el objetivo de los algoritmos de encaminamiento es la búsqueda de las rutas en una red que, satisfagan una serie de condiciones. Por ejemplo: rutas de mínimo coste económico, de mínimo retardo, de máxima cadencia eficaz o que satisfagan algún criterio administrativo.

Decisión de encaminamiento: Si la red es no orientada a conexión la decisión debe tomarse por cada datagrama. Si es orientada a conexión, únicamente durante el establecimiento del circuito virtual.

Cuando se establecen rutas para el transporte de los datos, los enrutadores deben calcular cuál es el camino óptimo y adicionarlo a sus tablas de enrutamiento, analizando parámetros como:

Coste o Distancia: es el coste que tiene el enlace entre enrutadores, algunos aspectos que se consideran para el cálculo del coste pueden ser: retardo, velocidad, precio, QoS, etc.



El coste puede ser dinámico: C=f(t)

Las conexiones entre routers pueden ser un enlace punto a punto, ATM, Frame Relay, Ethernet, etc.

La métrica es la magnitud a optimizar, podría ser retardo, cadencia (ancho de banda), coste económico, etc. Por lo general representa un solo parámetro pero podría ser más de uno. Está relacionado con el coste.

El valor de la métrica la deciden los enrutadores (aunque la configuración puede se hecha por los operadores), en todo caso la magnitud siempre deberá ser la misma para todos los enrutadores.

Eliminado: 89¶

Las *Tablas de encaminamiento*, que podrían ser estáticas o dinámicas, analizan, calculan y deciden el mejor camino hacia el destino. Esta información se encuentra en las tablas correspondientes a cada uno de los enrutadores.

Un ejemplo de tabla de encaminamiento para el enrutador A (Nodo A en la figura anterior), es la que se muestra a continuación.

Enrutador F		
Destino	**Siguiente**	**Coste**
A	A	0
B	B	2
C	B	5
D	D	6
E	B	3
F	B	4
G	B	7

Eliminado: 1

Eliminado: 2

2.1.1. Tipos de Encaminamiento

Para estudiar los tipos de encaminamiento, podríamos considerar dos maneras de realizar la subdivisión, ya sea analizando quién decide el camino a seguir y considerando su adaptabilidad. Para nuestro caso, si bien mencionaremos ambas, nos centraremos en el segundo aspecto.

Según QUIÉN DECIDE el camino a seguir:

Se pueden dividir en Fijado en el Origen y en Salto a Salto.

- Fijado en el Origen: el nodo origen decide el camino del paquete y ubica la información en la cabecera de dicho paquete (source routing bridges).

- Salto a Salto: cada uno de los enrutadores decide el salto que va a dar en base al destino almacenado en su tabla de encaminamiento.

Eliminado: 89¶

Según su ADAPTABILIDAD*:*

Si las Tablas de encaminamiento cambian a lo largo del tiempo. Se pueden dividir en Estáticos, Cuasi-estáticos y Dinámicos

- Estáticos: generalmente utilizados en redes pequeños. No hay cambios en las tablas de encaminamiento. Se pueden perder muchas tramas cuando cae un nodo.
- Dinámicos: Utilizado generalmente en redes grandes. Existen cambios en las tablas de encaminamiento. A su vez se subdividen en Centralizados, Aislados y Distribuidos.
  - *Centralizados*: Utilizados en redes X.25, existe un centro de control de red que toma las decisiones y conoce los caminos.
  - *Aislados*: se encaminan sin los nodos intercambiar información.
  - *Distribuidos*: No hay centro de control, los nodos hablan entre ellos y en base al cambio de información, crean las tablas.
- Cuasi-estáticos: son intermedios entre los estáticos y los dinámicos.

Eliminado: ¶

Eliminado: ¶

Eliminado: ¶

Eliminado: ¶

Eliminado: ¶

Eliminado: 1

Eliminado: 2

2.1.1.1. Encaminamiento Distribuido

Cada nodo intercambia información con otros nodos y a partir de ella calcula sus tablas de encaminamiento

Tipos de Encaminamiento Distribuido

Vector Distancia: En este tipo de encaminamiento, cada nodo “habla” solamente con sus vecinos y calcula las rutas a partir de la información suministrada por dichos vecinos topológicos (visión parcial del estado de la red).

Eliminado: ¶

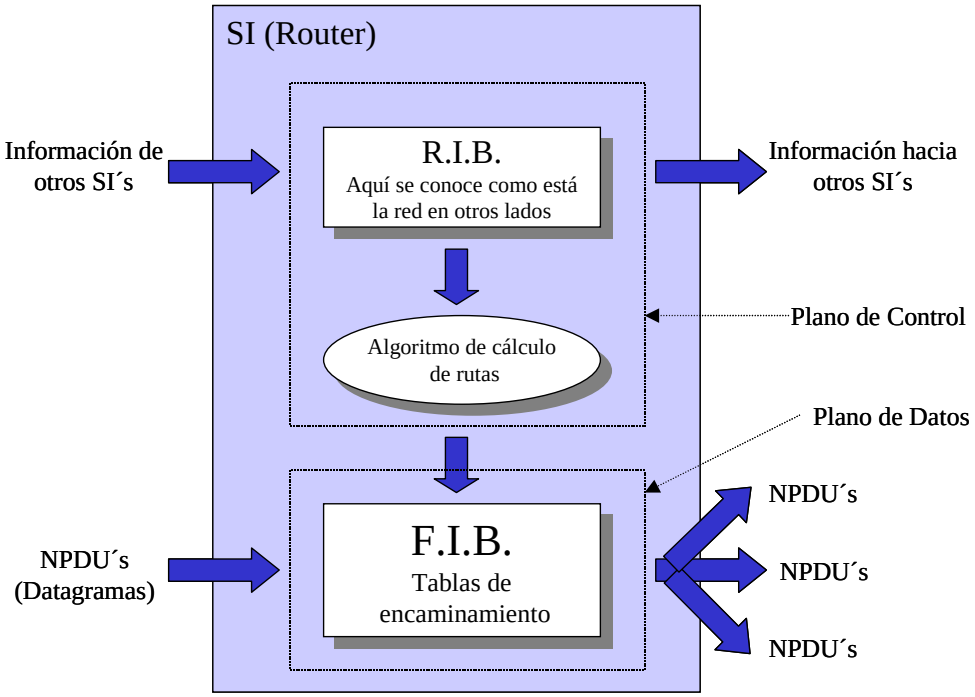
Estado de Enlaces: Cada informa al resto del estado de sus enlaces. Con la información recibida “construye un “mapa” completo de la red y sobre él ejecuta un algoritmo de

cálculo de rutas (visión global del estado de la red). Es un algoritmo más complejo que el anterior pero más usado por su optimalidad a pesar de que requiere mejores características en el hardware. En el gráfico siguiente se encuentra esquematizado cómo trabaja internamente un enrutador.

En primer lugar hay que dividir las funciones de un enrutador en dos planos:

El Plano de Datos (F.I.B. Forwarding Information Base) que es el encargado de mantener actualizadas las tablas de encaminamiento. Además su función es la de recibir los datagramas y encaminarlos hacia su destino por la ruta óptima, misma que se basa en la información obtenida desde el plano de control.

Plano de Control (R.I.B. Routing Information Base, y Algoritmos de cálculo de rutas) que es el que recibe la información de otros Sistemas Internos (SI) y mediante el algoritmo de enrutamiento calcula las rutas óptimas



Eliminado: 89¶

Eliminado: 1

Eliminado: 2

2.1.2. Encaminamiento Estático

Este tipo de algoritmos no basan sus decisiones de enrutamiento en mediciones o estimaciones del tráfico y la topología actuales. En cambio, la decisión de qué ruta se usará para llegar de origen a destino se calcula por adelantado, fuera de línea y se cargan en los enrutadores al iniciar la red.

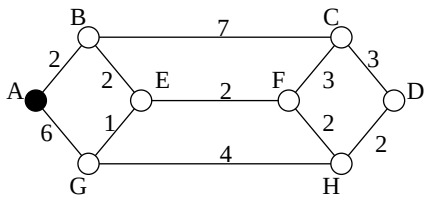
Entre los Algoritmos estáticos de enrutamiento tenemos:

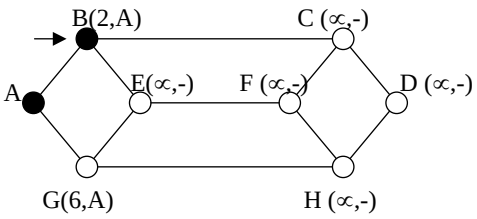
Enrutamiento por la trayectoria más corta

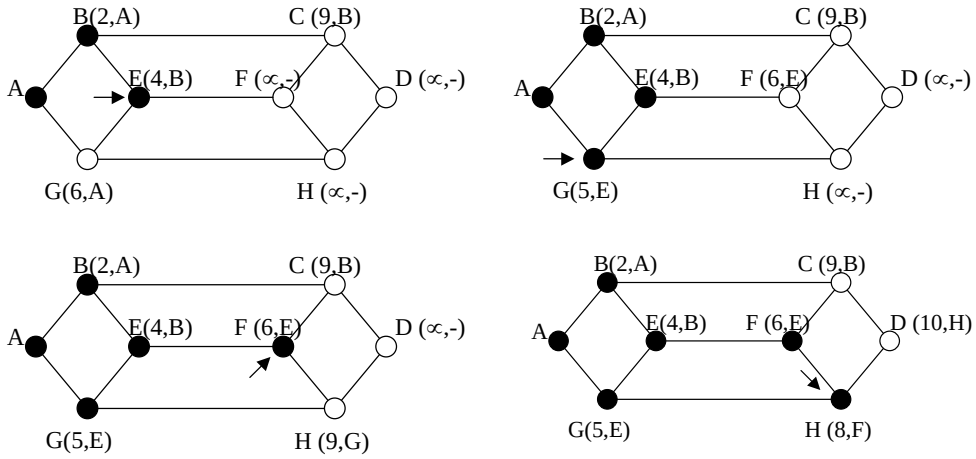
Es uno de los algoritmos de enrutamiento de mayor uso, tanto por su facilidad de entendimiento como por su sencillez. Para escoger una ruta entre dos enrutadores adyacentes, el algoritmo encuentra la trayectoria más corta entre ellos.

Una manera de medir la trayectoria entre dos enrutadores es por la cantidad de saltos, otra métrica es la distancia geográfica, además existen métricas como el retardo medio de encolamiento y transmisión de un paquete de prueba estándar, determinado por series de pruebas cada hora, así, la mejor trayectoria será la más rápida.

Podría también considerarse el ancho de banda, el tráfico medio, el costo de comunicación, la longitud media de las colas, etc. como métricas para establecer el trayecto.







Los primeros cinco pasos del cálculo de la trayectoria más corta de A a D

Inundación

En este tipo de algoritmo, cada paquete de entrada se envía por cada una de las líneas de salida, excepto por la que ingresó. En este algoritmo se generan grandes cantidades de paquetes duplicados que podría llegar a ser infinito. Para contrarrestar esto, se hace uso de un contador de saltos contenido en la cabecera de cada paquete, el cual disminuye en cada salto, descartándose el paquete al llegar el contador a cero. El número máximo ideal del contador debería ser la longitud de la trayectoria entre el origen y el destino. Si el transmisor no conoce el tamaño de la trayectoria, puede inicializar el contador al peor caso, es decir, al diámetro total de la subred.

Una técnica para evitar la inundación es llevar un registro de los paquetes diseminados, para evitar enviarlos una segunda vez. Para esto , el enrutador de origen pone un número de secuencia en cada paquete que recibe de sus hosts. Por tanto, cada enrutador necesita una lista por cada enrutador de origen que indique los números de secuencia originados en ese enrutador que ya ha visto. Si un paquete de entrada está en la lista, no se disemina.

Para evitar que la lista crezca sin límites, cada lista debe incluir un contador, *k*, que indique que todos los números de secuencia hasta *k* ya han sido vistos. Al llegar un

paquete, es fácil comprobar si el paquete es un duplicado; de ser así, se descarta. Es más, no se necesita la lista completa por debajo de *k*, pues *k* la resume efectivamente.

Una variación de la inundación, es la **inundación selectiva**; en este algoritmo los enrutadores no envían cada paquete de entrada por todas las líneas, sino solo por aquellas que van aproximadamente en la dirección correcta, a menos que la topología sea extremadamente peculiar.

La inundación no es muy práctica, excepto en algunos usos como las aplicaciones militares, donde podrían volar en pedazos varios enrutadores en cualquier momento. En bases de datos distribuidas donde es necesario actualizar concurrentemente todas las bases de datos. Otro posible uso sería como métrica contra la que pueden compararse otros algoritmos de enrutamiento ya que la inundación siempre escoge la trayectoria más corta posible.

Enrutamiento basado en el flujo

Los algoritmos estudiados hasta ahora solo toman en cuenta la topología; no consideran la carga. Si, por ejemplo, siempre hay una gran cantidad de tráfico entre dos enrutadores adyacentes, entonces podría resultar mejor encaminar el tráfico por una trayectoria alternativa que bien podría tener una mayor cantidad de saltos y de distancia geográfica, pero al mismo tiempo podría resultar mejor encaminarla por ahí.

El algoritmo de enrutamiento basado en el flujo considera tanto la topología como la carga para el enrutamiento. En algunas redes, la tasa media de flujo de datos entre cada par de nodos es relativamente estable y predecible. La idea en la que se basa el análisis es que, para una línea dada, se conocen la capacidad y el flujo promedio es posible calcular el retardo promedio de los paquetes en esa línea a partir de la teoría de colas. De los retardos promedio de todas las líneas, es directo el cálculo de un promedio ponderado por el flujo para obtener el retardo de paquete medio de la subred completa. El problema de enrutamiento se reduce a encontrar el algoritmo de enrutamiento que produzca el retardo promedio mínimo para la subred.

Eliminado: 89¶

Eliminado: ¶

Para utilizar esta técnica debe conocerse por adelantado cierta información. Primero, debe conocerse la topología de la subred. Segundo, debe estar dada la matriz de tráfico, Fij (trafico entre enrutadores destino y origen). Tercero, debe estar disponible la matriz de capacidad, Cij, donde se especifica la capacidad de cada línea en bps. Por último, debe escogerse un algoritmo tentativo de enrutamiento.

Eliminado: 1

Eliminado: 2

2.1.3. Encaminamiento Dinámico

Este tipo de algoritmos cambian sus decisiones de enrutamiento para reflejar los cambios de topología, y generalmente también el tráfico. Los algoritmos dinámicos difieren el lugar de obtención de su información (por ejemplo de los enrutadores adyacentes o de todos los enrutadores), el momento de cambio de sus rutas (por ejemplo cuando cambia la carga o cuando cambia la topología), y la métrica usada para la optimalidad (por ejemplo, distancia, número de saltos o tiempo de transferencia).

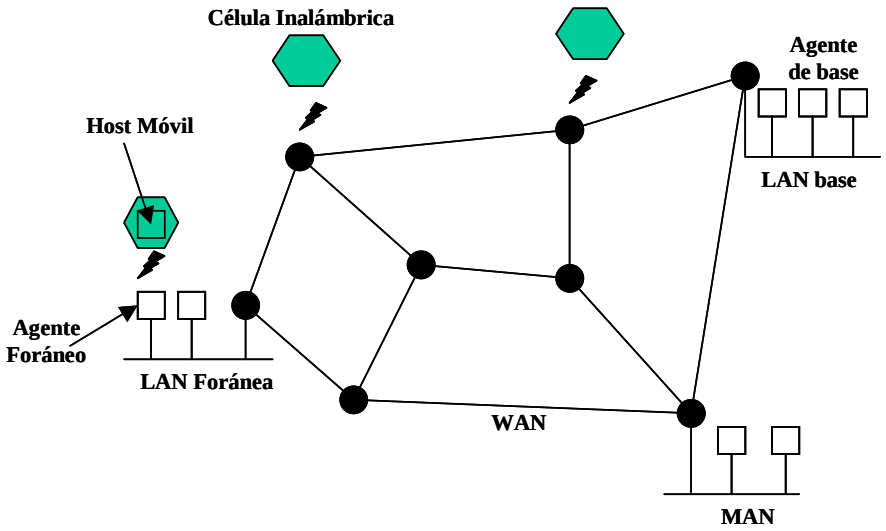
Entre los algoritmos de enrutamiento dinámico que revisaremos, tenemos: Enrutamiento por vector de distancia, Enrutamiento por estado del enlace, Enrutamiento jerárquico, Enrutamiento para hosts móviles, Enrutamiento por difusión, Enrutamiento por multitransmisión.

De la misma manera que los algoritmos de encaminamiento estáticos, a los dinámicos se los describirá brevemente excepto el de Vector distancia, estado de enlace y jerárquico que se los analizará más profundamente.

Enrutamiento para hosts móviles

Hoy día, millones de personas tienen computadoras portátiles, y generalmente quieren leer su correo electrónico y acceder a sus sistemas de archivos normales desde cualquier lugar del mundo. Estos *hosts* móviles generan una nueva complicación: para enrutar un paquete a un host móvil, la red primero tiene que encontrarlo.

Un ejemplo de cómo están diseñadas las redes actuales se muestra en la siguiente figura.



Aquí tenemos una WAN que consiste en enrutadores y *host*. Conectadas a la WAN hay varias LAN y MAN y células inalámbricas.

Se supone que todos los usuarios poseen una **localidad base** que nunca cambia. Los usuarios también tienen una dirección base, que puede servir para localizar su localidad base. La meta de enrutamiento en los sistemas con usuarios móviles es posibilitar el envío de paquetes a usuarios móviles usando su dirección base, y hacer que los paquetes lleguen eficientemente a ellos en cualquier lugar en el que puedan estar. Lo difícil es encontrarlos.

El mundo podría estar dividido (geográficamente) en unidades pequeñas, que podrían denominarse áreas, siendo un área típicamente una LAN o una célula inalámbrica. Cada área tiene uno o más agentes foráneos, que llevan el registro de todos los usuarios que visitan el área. Además, cada área tiene un agente de base, que lleva el registro de todos los usuarios móviles cuya base está en el área, pero que actualmente están visitando otro área.

Al entrar un usuario nuevo en un área, ya sea al conectarse a ella, o simplemente al entrar en la célula, su computadora debe registrarse con el agente foráneo de ese lugar. El procedimiento de registro funciona de la siguiente manera:

Eliminado: 89¶

Eliminado: ¶

a) Los agentes foráneos detectan la presencia de un host móvil, ya sea a través de un paquete que difunde (periódicamente) indicando su presencia y dirección o a través de un paquete que envía el host preguntando si existe un agente foráneo (cuando entra en el área).
b) El host móvil se registra con el agente foráneo, dando su dirección base, su dirección actual de capa de enlace de datos y cierta información de seguridad.
c) El agente foráneo avisa al agente base del host móvil que éste se encuentra en el área del agente foráneo.
d) El agente base examina la información de seguridad que le envió el agente foráneo para establecer si fue generada en los últimos segundos e indicarle que proceda.
e) Cuando el agente foráneo recibe el reconocimiento del agente base, hace una entrada en sus tablas e informa al host móvil que ahora esta registrado

Cuando el usuario sale del área debe anunciarlo para eliminarlo del registro.

Cuando un paquete es enviado al usuario móvil, se enruta a la LAN base del usuario, este es captado por su agente base, el mismo que encapsula el paquete en el campo de carga útil de un paquete exterior (tuneling) y lo envía al agente foráneo, el mismo que lo desencapsula y lo envía al usuario móvil.

En segundo lugar el agente base indica al transmisor que envíe los paquetes haciendo tuneling, dirigidos al agente foráneo para que puedan llegar directamente al host móvil.

Eliminado: ¶

Eliminado: ---Salto de página---

Enrutamiento por difusión

En algunas aplicaciones, los host necesitan enviar mensajes a varios otros host o a todos los demás. El envío simultáneo de paquetes a todos los destinos se llama **difusión**, existen varias técnicas para llevarla a cabo.

Eliminado: ¶

- Una técnica que no requiere características especiales es que el origen envía copias del destino a todos los paquetes. Esta técnica desperdicia mucho ancho de banda y requiere que el origen mantenga una lista completa de todos los destinos.

Eliminado: 89¶

- Otra técnica es la inundación, que se analizó anteriormente.
- Un tercer algoritmo es el enrutamiento multidestino. En este método cada paquete contiene un listado de destinos o un mapa de bits que indica los destinos deseados. Al llegar un paquete al enrutador, este revisa todos los destinos para determinar el grupo de líneas de salida que necesitará. El enrutador genera una copia nueva del paquete para cada línea de salida a usar, e incluye en cada paquete solo aquellos destinos que usan la línea. Tras una cantidad suficiente de saltos, cada paquete llevará solo un destino.
- Otro algoritmo de difusión usa el árbol de descenso para el enrutador que inicia la difusión, o cualquier otro árbol de extensión adecuado. El árbol de extensión es un subgrupo de la subred que incluye todos los enrutadores pero no contiene ciclos. Si cada enrutador sabe cuáles de sus líneas pertenecen al árbol de extensión, puede copiar un paquete de entrada difundido en todas las líneas del árbol de extensión, excepto en aquella por la que llegó. Este método hace un uso excelente del ancho de banda, generando una cantidad mínima de paquetes necesarios para llevar a cabo el trabajo. El inconveniente es que cada enrutador debe conocer algún árbol de extensión para que pueda funcionar.
- El último algoritmo de difusión es un intento de aproximar el comportamiento del anterior aún cuando los enrutadores no saben nada sobre árboles de extensión. Cuando llega un paquete difundido a un enrutador, éste lo revisa para ver si llegá por la línea normalmente usada para enviar paquetes al origen de difusión. De ser así, es posible que el paquete difundido haya seguido la mejor ruta desde el enrutador, y por tanto, sea la primera copia en llegar al enrutador. Siendo éste el caso, en enrutador reenvía copias del paquete por todas las líneas, excpeto por la que llegó. Sin embargo, si el paquete difundido llegó por una línea diferente de la preferida para llegar al origen, se descarta el paquete como probable duplicado. A este algoritmo se lo denomina *reenvío por trayectoria invertida.*

Enrutamiento por multitransmisión

Este tipo de algoritmo es utilizado cuando se necesita enviar información a un grupo de la red y no a la totalidad de ella. Para esto, cada enrutador calcula un árbol de extensión que

Eliminado: 89¶

cubre a todos los demás enrutadores de la subred. Por ejemplo en la figura a continuación (a) tenemos una subred con dos grupos, 1 y 2. Algunos enrutadores están conectados a hosts que pertenecen a uno o ambos grupos. En la figura (b) se muestra un árbol de extensión para el enrutador de la izquierda. Al enviar un proceso un paquete multitransmisión a un grupo, el primer enrutador examina su árbol de extensión y lo recorta, removiendo todas las líneas que no conducen a hosts que son miembros del grupo. En la figura (c), se muestra el árbol de extensión recortado del grupo 1. En (d) se presenta el árbol de extensión recortado del grupo 2. Los paquetes multitransmisión se reenvían solo a través del árbol de extensión apropiado.

Eliminado: ¶
¶

Hay varias maneras para recortar el árbol de extensión. La más sencilla puede usarse si se maneja enrutamiento por estado de enlace, y cada enrutador está consciente de la topología completa de la subred, incluyendo qué hosts pertenecen a cuáles grupos. Entonces puede recortarse el árbol comenzando por el final de cada trayectoria y trabajando hacia la raíz, removiendo todos los enrutadores que no pertenecen al grupo en cuestión. Con el enrutador por vector distancia el algoritmo básico es el envío por trayectoria invertida.

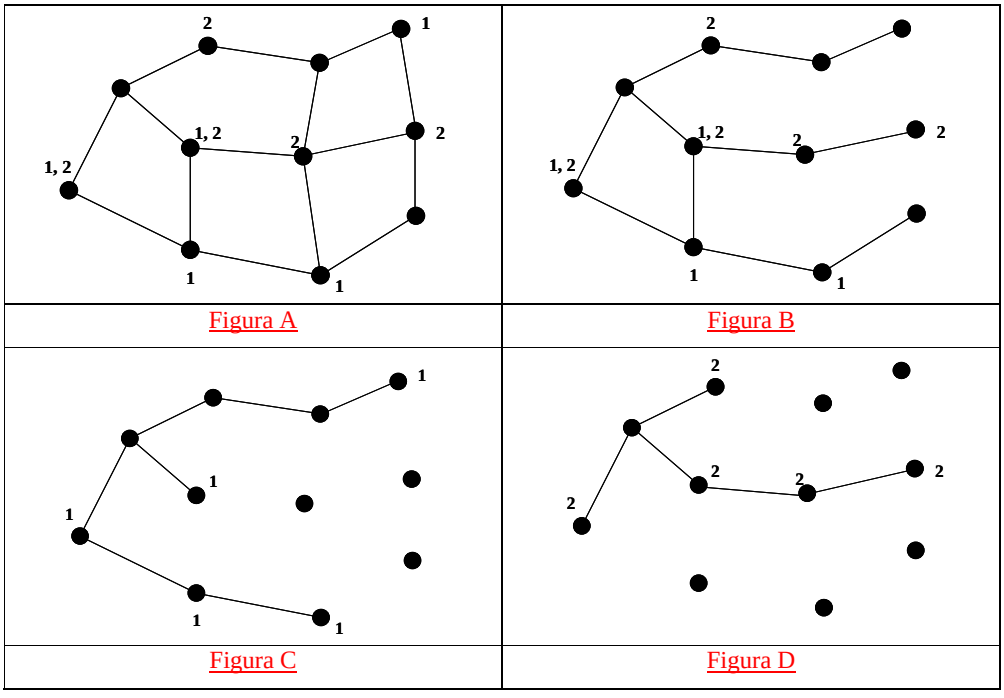


Figura A | Figura B | Figura C | Figura D

Eliminado: 89¶

Eliminado: *1*

Eliminado: *3*

2.2. Algoritmo de Vector Distancia

A continuación revisaremos detalladamente uno de los algoritmos dinámicos más utilizados para el encaminamiento de la información. En este tipo de algoritmo, el enrutador guarda en su tabla de encaminamiento la mejor distancia y el mejor camino para llegar a un destino determinado. Estas tablas se actualizan intercambiando información con los enrutadores vecinos. En la tabla de encaminamiento, cada registro contiene información de cada uno de los enrutadores.

Eliminado: ¶

Cada registro está dividido en dos partes: la primera contiene la mejor métrica y la segunda el mejor camino hacia el destino. Esta métrica podría ser: distancia física, cantidad de escalas, tiempo de retardo, cantidad de flujo , etc. Cada enrutador conoce "la distancia" a cada uno de los enrutadores vecinos. Si esta métrica es de escalas, la distancia simplemente es una escala. Si la métrica es longitud de la cola, el enrutador examina cada cola. Si la métrica es el retardo, el enrutador puede hacer uso de paquetes de ECO y detectar el tiempo de retardo del paquete para poder establecer este parámetro.

Analizaremos el siguiente ejemplo: supongamos que se utiliza como métrica el retardo y que el enrutador en análisis conoce el retardo a cada uno de sus vecinos. Una vez cada *T mseg.*, cada enrutador envía a todos sus vecinos una lista de sus retardos estimados a cada destino.

También recibe una lista similar de cada vecino. Imagine que una de estas tablas acaba de llegar de un vecino *X*, siendo *Xi*, la estimación de *X* respecto al tiempo que le toma llegar al enrutador *i*. Si el enrutador sabe que el retardo a *X* es de *m mseg.*, también sabe que puede alcanzar el enrutador *i* a través de *X* en *Xi + m mseg.*vía *X*. Efectuando este cálculo para cada vecino, un enrutador puede encontrar la estimación que parezca ser la mejor y usar esa estimación y la línea correspondiente en su nueva tabla de enrutamiento. Nótese que la vieja tabla de enrutamiento no se usa en este cálculo.

Este proceso de actualización se puede apreciar en la figura de la siguiente página. En la parte izquierda se muestra la subred a considerar. En las cuatro primeros columnas de la

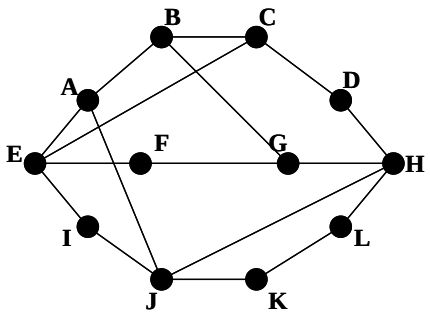
Eliminado: 89¶

derecha 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. Supóngase que J a medido o estimado el retardo a sus miembros, *A, I, H* y *K*, en 8, 10, 12 y 6 mseg. respectivamente. Cuando *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.

Eliminado: ¶
¶

El mejor de estos valores es 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 a usar 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 la parte derecha.

Eliminado: ¶



Nuevo retardo desde J

A	A	I	H	K	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

Vectores recibidos de los cuatro vecinos de J

Nueva tabla para J

JA=8 JI=10 JH=12 JK=6

Eliminado: ¶

Otro ejemplo que nos permite analizar paso por paso cómo se van generando las tablas de enrutamiento en cada uno de los nodos que intervienen en una subred, se muestra en las figuras de la siguiente página.

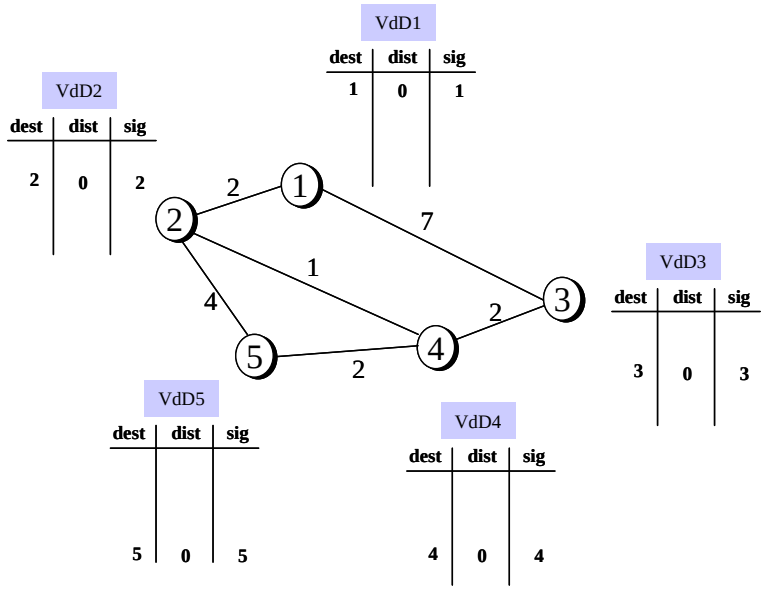
Eliminado: siguientes

Eliminado: ¶

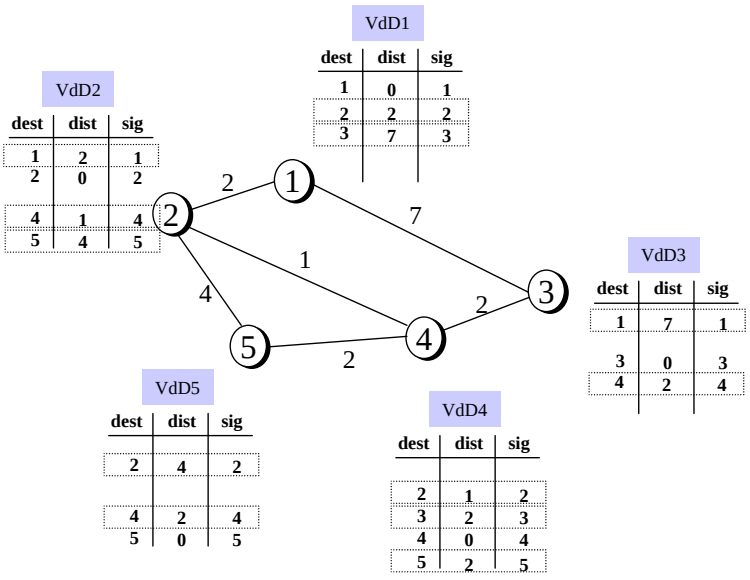
Eliminado: 89¶

Eliminado:)

En el primer gráfico (Estado inicial), se observa que cada uno de los enrutadores (nodos) tiene en su tabla únicamente el valor que corresponde a la distancia a si mismo, es decir 0.

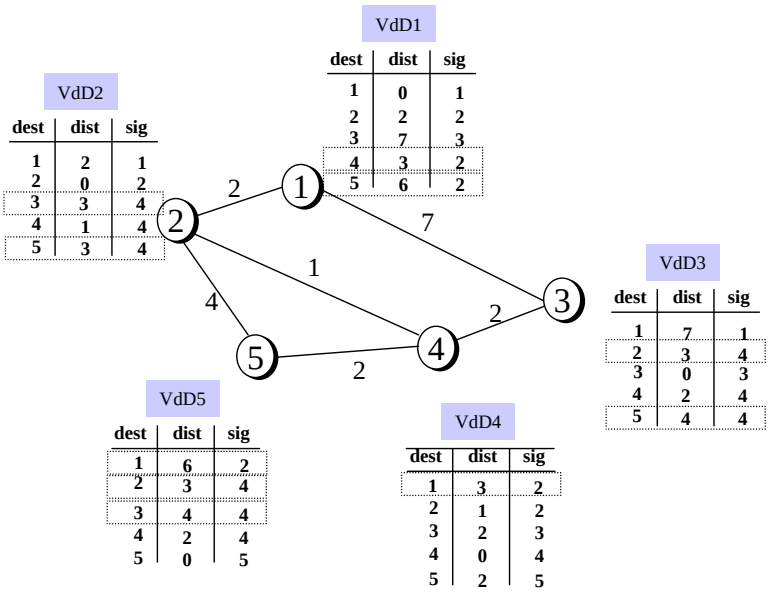


En el primer paso (gráfico inferior) cada enrutador transmite y recibe la métrica de cada uno de sus vecinos, es decir, de sus distancias, y las añade a su propia tabla.

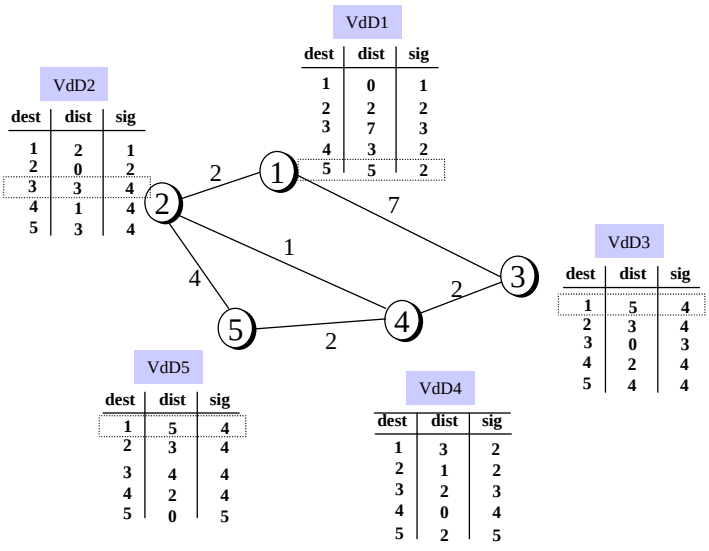


Eliminado: 89¶

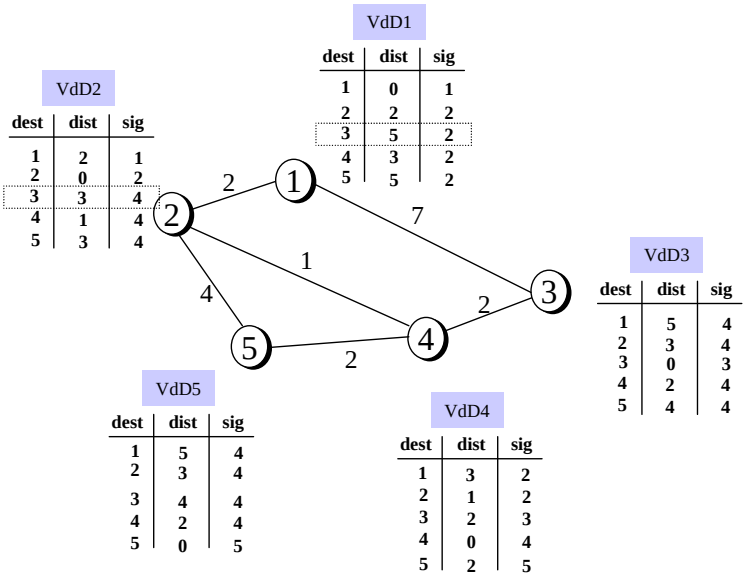
En el segundo paso (siguiente gráfico), los enrutadores reciben la información almacenada de sus vecinos y compara con su información, luego ingresar en su tabla los datos que permiten obtener distancias más cortas a los enrutadores



En el tercer paso los enrutadores reciben la información de las tablas de enrutamiento de sus vecinos y las agregan a sus propias tablas. Cuando esta información se refiere a datos ya existentes, la comparan y escogen la mejor opción de métrica.



Finalmente las tablas de los enrutadores quedan armadas en base a la mejor información existente de métrica.



Cada vez que existen cambios en la subred, los enrutadores envían información la cual será actualizada en cada uno de ellos. Las entradas siempre se refrescan, si una información deja de llegar, esa ruta es borrada de las tablas.

Podríamos resumir las características más importantes del algoritmo por vector de distancia de la siguiente manera:

Ventajas:

- Muy sencillo: pocas líneas de código
- Robusto (simple de implementar, fallos conocidos)
- Tablas pequeñas: Consume poco CPU, solo recibe datos y compara. (RIB = VdD de los vecinos únicamente)

Desventajas:

- Convergencia lenta: pueden aparecer bucles
- Crecimiento difícil

Eliminado: 89¶

Eliminado: ¶

Mecanismos correctores:

- Trigered updates: no se actualiza cada determinado tiempo (30 seg) sino solamente cuando hay cambios
- Split-horizon: incluye las rutas en las actualizaciones enviadas al enrutador el que se aprendieron, pero pone sus métricas a infinito.
- Poisson reversed: transmite rutas negativas cuando "cae" un nodo, para que lo puedan borrar de las tablas.

Protocolos que utilizan el algoritmo de Vector Distancia:

- RIP,
- HELLO,
- IGRP,
- EIGRP.

Eliminado: 1

Eliminado: 3

2.2.1. El problema del conteo a infinito

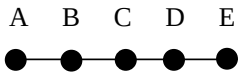
El enrutamiento por vector de distancia funciona en teoría, pero tiene un problema serio en la práctica: aunque converge en la respuesta correcta, puede hacerlo lentamente. En particular, reacciona con rapidez a las buenas noticias, pero con lentitud ante las malas. Considere un enrutador cuya mejor ruta al destino X es larga. Si en el siguiente intercambio el vecino A informa repentinamente un retardo corto a X, el enrutador simplemente se conmuta a modo de usar la línea a A para enviar tráfico hasta X. En un intercambio de vectores, se procesan las buenas noticias.

Para ver la rapidez de propagación de las buenas noticias, considere las subred de 3 nodos (lineal) de la siguiente figura, en la que la métrica de retardo es el número de saltos. Supóngase que *A* está desactivado inicialmente y que los otros enrutadores lo saben. Es decir, habrán registrado como infinito el retardo a *A*.

Al activarse A, los demás enrutadores saben de él, gracias a los intercambios de vectores. Dado que existe un intercambio de información, en el primer intercambio , *B* se entera de que su vecino de la izquierda tiene un retardo de 0 hacia *A*. *B* crea entonces una entrada en

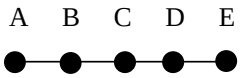
Eliminado: 89¶

su tabla de enrutamiento, indicando que *A* está a un salto de distancia hacia la izquierda. Los demás enrutadores aún piensan que *A* está desactivado. Las entradas de las tablas de enrutamiento de *A* en este punto tiene un valor de 1 para B y para el resto de enrutadores es ∝ (Figura A). Durante el siguiente intercambio, *C* se entera de que *B* tiene una trayectoria a *A* de longitud 1, por lo que actualiza su tabla de enrutamiento para indicar una trayectoria de longitud 2, pero *D* y *E* no se enteran de las buenas nuevas sino hasta después. Como se puede observar, las buenas noticias se difunden a razón de una escala por intercambio. En una subred cuya trayectoria mayor tiene una longitud de *N* escalas, en un lapso de *N* intercambios todos los enrutadores sabrán sobre los enlaces y los enrutadores recientemente revividos.



∝	∝	∝	∝	Inicialmente
1	∝	∝	∝	Tras 1 intercambio
1	2	∝	∝	Tras 2 intercambios
1	2	3	∝	Tras 3 intercambios
1	2	3	4	Tras 4 intercambios

Figura A



1	2	3	4	Inicialmente
3	2	3	4	Tras 1 intercambio
3	4	3	4	Tras 2 intercambios
5	4	5	4	Tras 3 intercambios
5	6	5	6	Tras 4 intercambios
7	6	7	6	Tras 5 intercambios
7	8	7	8	Tras 6 intercambios

Eliminado: ¶

∝ ∝ ∝ ∝

Figura B

Ahora consideraremos la situación en la que todas los enlaces y enrutadores están activos inicialmente (Figura B). Los enrutadores *B*, *C*, *D* y *E* tienen distancias a *A* de 1, 2, 3 y 4,

Eliminado: 89¶

respectivamente. De pronto *A* se desactiva, o se corta el enlace entre *A* y *B*, que es lo mismo desde el punto de vista de *B*.

En el primer intercambio de paquetes, *B* no escucha nada de *A*. Pero, *C* dice: "hay una trayectoria a *A* de longitud 2". *B* no sabe que la trayectoria de *C* pasa a través de *B* mismo. Hasta donde *B* sabe, *C* puede tener 10 enlaces de salida, todas con trayectorias independientes a *A* de longitud 2. Como resultado, *B* ahora piensa que puede llegar a *A* por medio de *C*, con una longitud de trayectoria de 3. *D* y *E* no actualizan sus tablas de enrutamiento para *A* en el primer intercambio.

En el segundo intercambio, *C* nota que cada uno de sus vecinos indica tener una trayectoria a *A* de longitud 3. *C* escoge una de ellas al azar y hace que su nueva distancia a *A* sea de 4. Los intercambios subsecuentes se muestran en la figura B.

A partir de esta figura, queda clara la razón por la que las malas noticias viajan con lentitud: ningún enrutador tiene jamás un valor mayor en más de una unidad que el mínimo de todos sus vecinos. Gradualmente, todos los enrutadores elevan sus cuentas hacia el infinito. Por esta razón, es prudente hacer infinito igual a la trayectoria más larga, más 1. Si la métrica es el retardo de tiempo, no hay un límite superior bien definido, por lo que se necesita un valor alto para evitar tratar una trayectoria con un retardo grande como si estuviera desactivada. Este problema es conocido como el de **conteo al infinito**.

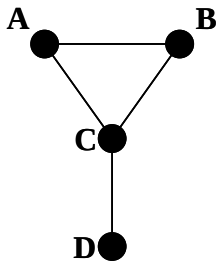
Eliminado: 1
Eliminado: 3

2.2.2. Recorte por horizonte dividido (split horizon)

Existen algunos métodos para contrarrestar el problema del conteo al infinito, a continuación se describe uno de ellos y se indica también porqué falla. El algoritmo de **horizonte dividido** funciona de la misma manera que el enrutamiento por vector distancia, excepto que la distancia a *X* no se informa en el enlace por el que se envían paquetes para *X* (en realidad, se informa como infinita). En el estado inicial de la figura anterior, por ejemplo, *C* "le dice" a *D* "la verdad" sobre la distancia a *A*, pero *C* "le dice" a

B que la distancia a *A* es infinita. De la misma manera, *D* "le dice" la verdad a *E*, pero "le miente" a *C*.

Veamos ahora lo que pasa cuando *A* se desactiva. En el primer intercambio, *B* descubre que el enlace directo desapareció, y *C* está informando también una distancia infinita a A. Dado que ninguno de sus vecinos puede llegar a *A*, *B* establece también si distancia como infinita. En el siguiente intercambio, *C* escucha que *A* es inalcanzable desde sus dos vecinos, por lo que también marca a *A* como inalcanzable. Usando el horizonte dividido, las malas noticias se propagan a razón de una escala por intercambio. Esta velocidad es mucho mejor que sin el horizonte dividido.



Ejemplo en donde falla el Horizonte dividido

La verdadera mala noticia es que el horizonte dividido, aunque se utiliza ampliamente, a veces falla. Considere, por ejemplo, la subred de cuatro nodos de la figura siguiente. Inicialmente, tanto *A* como *B* tienen una distancia a *D* de 2, y *C* tiene una distancia de 1 al mismo lugar.

Ahora suponga que se desactiva el enlace *CD*. Usando el horizonte dividido, tanto *A* como *B* le indican a *C* que no puede llegar a *D*. Por tanto, *C* concluye de inmediato que *D* es inalcanzable y lo informa tanto a *A* como a *B*. Desafortunadamente *A* escucha que *B* tiene una trayectoria de longitud 2 a *D*, por lo que supone que puede llegar a *D* a través de *B* en tres escalas. De manera parecida, *B* concluye que puede llegar a *D* en tres escalas a través de *A*. En el siguiente intercambio, cada uno establece en 4 su distancia a *D*. Ambos cuentan gradualmente a infinito, que es precisamente el comportamiento que estamos tratando de evitar.

Eliminado: 89¶

2.3. Algoritmo de Estado de Enlaces

Eliminado: ¶
¶
Eliminado: 1
Eliminado: 4

El enrutamiento por vector de distancia se usó en ARPANET hasta 1979, cuando fue reemplazado por el enrutamiento por estado de enlace. Dos problemas principales causaron su defunción. Primero, dado que la métrica de retardo era la longitud de la cola, no tomaba en cuenta el ancho de banda al escoger rutas. Inicialmente, todas las líneas eran de 56 kbps, por lo que el ancho de banda no era importante, pero una vez que se modernizaron algunas líneas a 230 kbps y otras a 1,544 Mbps, el no tomar en cuenta el ancho de banda se volvió un problema importante. Por supuesto, habría sido posible cambiar la métrica de retardo para considerar también el ancho de banda, pero existía un segundo problema, que el algoritmo con frecuencia tardaba demasiado en convergir, aún con trucos como el horizonte dividido. Por estas razones el algoritmo fue reemplazado por uno completamente nuevo, llamado **enrutamiento por estado de enlace.** Hoy en día se utilizan algunas variantes de este tipo de enrutamiento.

El concepto en que se basa el enrutamiento por estado de enlaces es sencillo y puede definirse en cinco partes, en las cuales cada enrutador debe:

- Descubrir a sus vecinos y conocer sus direcciones de red.
- Medir el retardo o costo para cada uno de sus vecinos.
- Construir un paquete que indique todo lo que acaba de aprender.
- Enviar este paquete a todos los demás enrutadores.
- Calcular la trayectoria más corta a todos los demás enrutadores.

De hecho, la topología total y todos los retardos se miden experimentalmente y se distribuyen a cada enrutador. Entonces puede usarse el algoritmo de Dijkstra (ver Anexo 1) para encontrar la trayectoria más corta a todos los demás enrutadores.

Eliminado: 89¶

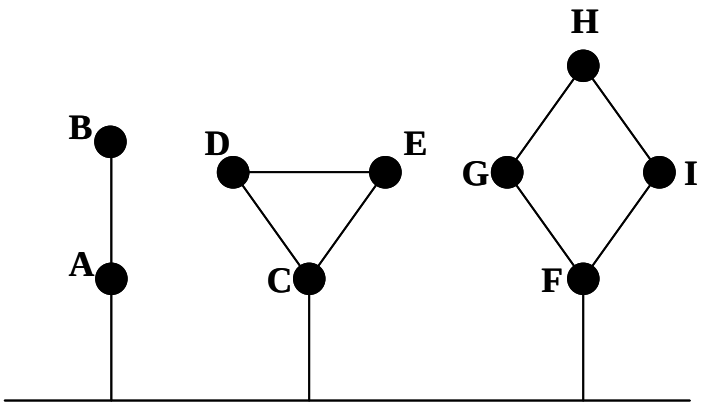
Eliminado: 1

Eliminado: 4

2.3.1. Conocimiento de los vecinos

Al ponerse en operación un enrutador, su primera tarea es averiguar quienes son sus vecinos; esto lo logra enviando un paquete especial de “HELLO”, por cada línea punto a punto. Se espera que el enrutador del otro extremo envíe de regreso una respuesta indicando quién es. Estos nombres deben ser globalmente únicos puesto que, cuando un enrutador distante escucha después que tres enrutadores están conectados a **F**, es indispensable que pueda determinar si los tres se refieren a este mismo enrutador *F*.

Al conectarse dos o más enrutadores mediante una LAN, la situación es más complicada. En la siguiente figura se muestra una LAN a la que están conectados directamente tres enrutadores, *A*, *C* y *F*. Cada uno de estos enrutadores está conectado a uno o más enrutadores adicionales.



Nueve enrutadores conectados a una LAN

Podría ser considerada la LAN como un nodo adicional *N* con el fin de definir trayectorias, por ejemplo para ir de *A* a *C* la trayectoria sería *ACN*.

Eliminado: 89¶

Eliminado: 1

Eliminado: 4

2.3.2. Medición del costo de la Línea

El algoritmo de enrutamiento por estado de enlace requiere que cada enrutador sepa, o al menos tenga una idea, del retardo a cada uno de sus vecinos. La manera más directa de determinar este retardo es enviar un paquete especial "ECHO", a través de la línea, el cual debe enviar de regreso inmediatamente el otro lado. Si mide el tiempo de ida y vuelta y lo divide entre dos, el enrutador transmisor puede tener una idea del retardo. Para obtener mejores resultados aún, la prueba puede llevarse a cabo varias veces y usarse el promedio.

Eliminado: ECO (

Eliminado: ver Anexo 3)

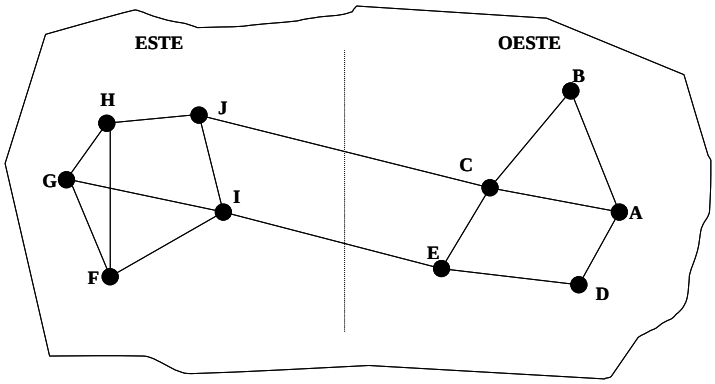
Un asunto interesante es si se debe tomar en cuenta la carga al medir el retardo. Para considerar la carga, el temporizador de viaje redondo debe iniciarse al encolar el paquete de ECO. Si se desea ignorar la carga, el temporizador debe iniciarse cuando el paquete de ECO llega al frente de la cola.

Pueden citarse argumentos a favor en ambos sentidos. La inclusión de los retardos inducidos por el tráfico en las mediciones implica que, cuando un enrutador puede escoger entre dos líneas con el mismo ancho de banda, una con carga alta continua y otra sin ella, considerará como trayectoria más corta la ruta a través de la línea sin carga. Esta selección resultará en un mejor desempeño.

Desafortunadamente, también hay un argumento en contra de la inclusión de la carga en el cálculo del retardo. Considere la subred de la siguiente figura, dividida en dos partes, este y oeste, conectadas por dos enlaces, *CF* y *EI*. Suponga que la mayor parte del tráfico entre el este y el oeste, usa el enlace *CF* y, como resultado, esta línea tiene tráfico alto con retardos grandes. La inclusión del retardo por encolamiento en el cálculo de la trayectoria más corta hará más atractiva a *EI*. Una vez instaladas las nuevas tablas de enrutamiento, la mayor parte del tráfico este-oeste, pasará ahora por *EI*. En consecuencia, en la siguiente actualización, *CF* aparecerá como la trayectoria más corta. Como resultado, las tablas de enrutamiento pueden oscilar sin control, conduciendo a un enrutamiento errático y muchos problemas potenciales. Si se ignora la carga y solo se considera el ancho de

Eliminado: 89¶

banda, no ocurre este problema. De manera alterna, puede distribuirse la carga entre ambas líneas, pero esta solución no aprovecha al máximo la mejor tayectoria.



Dos subredes conectadas por dos enlaces

Eliminado: 1

Eliminado: 4

2.3.3. Construcción de los paquetes de estado de enlaces

Una vez que se ha recabado la información necesaria para el intercambio, el siguiente paso es que cada enrutador construya un paquete con todos los datos. El paquete comienza con la identidad del transmisor, seguida de un número de secuencia, una edad (que se describirá después) y una lista de vecinos. Para cada vecino, se cita el retardo a ese vecino. En la figura A, se da un ejemplo de subred, mostrándose el retardo en las líneas. Los paquetes de estado de enlace de los seis enrutadores se muestran en la segunda figura B, a continuación.

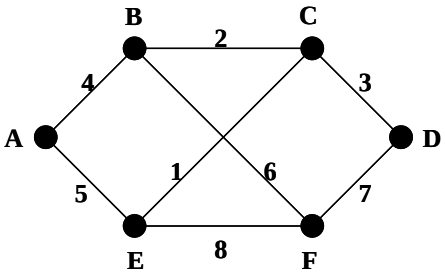


Figura A. Subred

Eliminado: 89¶

Enlace Estado Paquetes

A	
Sec.	
Edad	
B	4
E	5

B	
Sec.	
Edad	
A	4
C	2
F	6

C	
Sec.	
Edad	
B	2
D	3
E	1

D	
Sec.	
Edad	
C	3
F	7

E	
Sec.	
Edad	
A	5
C	1
F	8

F	
Sec.	
Edad	
B	6
D	7
E	8

Figura B. Paquetes de estado de enlace para esta subred

Es fácil construir los paquetes de estado de enlace. La parte difícil es determinar cuándo construirlos. Una posibilidad es construirlos periódicamente, es decir, a intervalos regulares. Otra posibilidad es al ocurrir un evento significativo, como la caída o reactivación de una línea o de un vecino, o el cambio apreciable de sus propiedades.

Eliminado: 1

Eliminado: 4

2.3.4. Distribución de los paquetes de estado de enlaces

La parte más complicada del algoritmo es la distribución confiable de los paquetes de estado de enlace. A medida que se distribuyen e instalan los paquetes, los enrutadores que reciban los primeros cambiarán sus rutas. En consecuencia, los distintos enrutadores podrían estar usando versiones diferentes de la topología, lo que puede conducir a inconsistencias, ciclos, máquinas inalcanzables y otros problemas.

Primero describiremos el algoritmo básico de distribución y luego lo refinaremos. La idea fundamental es usar inundación para distribuir los paquetes de estado de enlace. A fin de mantener controlada la inundación, cada paquete contiene un número de secuencia que se incrementa con cada paquete nuevo enviado. Los enrutadores llevan el registro de todos los pares (enrutador de origen, secuencia) que ven. Al llegar un paquete de estado de enlace, se revisa contra la lista de paquetes ya vistos. Si es nuevo, se reenvía a través de todas las líneas, excepto aquella por la que llegó. Si es un duplicado, se descarta. Si llega un paquete con número de secuencia menor que el mayor visto hasta el momento, se rechaza como obsoleto.

Este algoritmo tiene algunos problemas, pero son manejables. Primero, si los números de secuencia vuelven a comenzar, reinará la confusión. La solución aquí es usar un número de secuencia de 32 bits. Con un paquete de estado de enlace por segundo, el tiempo para volver a empezar será de 37 años, por lo que puede ignorar esta posibilidad.

Segundo, si llega a caerse un enrutador, perderá el registro de su número de secuencia. Si comienza nuevamente en 0, se rechazará como duplicado el siguiente paquete.

Tercero, si llega a corromperse un número de secuencia y se recibe 65540 en lugar de 4 (un error de 1 bit), los paquetes 5 a 65540 serán rechazados como obsoletos, dado que se piensa que el número de secuencia actual es 65540.

La solución a todos estos problemas es incluir la edad de cada paquete después del número de secuencia y disminuirla una vez cada segundo. Al llegar la edad a 0, se descarta la información de ese enrutador. Normalmente, entra un paquete nuevo cada 10 minutos, digamos, por lo que la información de los enrutadores solo expira cuando está caído el enrutador (o se pierden 6 paquetes consecutivos, evento poco probable). El campo de edad es también disminuido por cada enrutador durante el proceso inicial de inundación para asegurar que no pueda perderse ningún paquete y sobrevivir durante un período de tiempo indefinido (se descarta un paquete cuya edad es cero).

Algunos refinamientos de este algoritmo lo hacen más robusto. Al llegar un paquete de estado de enlace a un enrutador para ser inundado, no se encola para transmisión inmediata. En cambio, entra en un área de retención donde espera un tiempo corto. Si antes de transmitirlo entra otro paquete de estado de enlace de la misma fuente, se comparan sus números de secuencia. Si son iguales, se descarta el duplicado. Si son diferentes, se desecha el más viejo. Como protección contra los errores en los enlaces enrutador-enrutador, todos los paquetes del estado de enlace requieren reconocimiento. Al desactivarse un enlace, se examina el área de retención en orden por turno circular para seleccionar un paquete o reconocimiento a enviar.

La estructura de datos usada por el enrutador *B* para la subred de la siguiente figura anterior (figura A) se describe en la siguiente tabla. Cada renglón aquí corresponde a un paquete de estado de enlace recién llegado, pero aún no procesado por completo. La tabla registra dónde se originó el paquete, el número de secuencia y edad, y los datos. Además hay indicadores de transmisión y reconocimiento para cada uno de los tres enlaces de *B* (a *A*, *C* y *F* respectivamente). Los indicadores de envío significan que el paquete debe enviarse a través de la línea indicada. Los indicadores de reconocimiento significan que deben reconocerse ahí.

			Indicadores envío			Indicadores ACK			
Origen	Sec.	Edad	A	C	F	A	C	F	Datos
A	21	60	0	1	1	1	0	0	
F	21	60	1	1	0	0	0	1	
E	21	59	0	1	0	1	0	1	
C	20	60	1	0	1	0	1	0	
D	21	59	1	0	0	0	1	1	

Buffer de paquetes para el enrutador *B* de la figura A

En la tabla se puede observar que el paquete de estado de enlace *A* llegó directamente, por lo que debe enviarse a *F* y *C* y reconocerse ante *A*, como lo muestran los bits de indicación. De la misma manera, el paquete de *F* tiene que reenviarse a *A* y a *C*, y reconocerse ante *F*.

Sin embargo, la situación del tercer paquete, de *E*, es diferente; llegó dos veces, la primera a través de *EAB* y la segunda por medio de *EFB*. En consecuencia, este paquete tiene que enviarse sólo a *C*, pero reconocerse tanto ante *A* como ante *F*, como lo indican los bits.

Si llega un duplicado mientras el original aún está en el buffer, los bits tienen que cambiar. Por ejemplo, si llega una copia del estado de *C* desde *F* antes de reenviarse la cuarta entrada de la tabla, cambiarán los seis bits a 100011 para indicar que el paquete debe ser reconocido ante *F*, pero no enviarse ahí.

Eliminado: 89¶

Eliminado: 1

Eliminado: 4

2.3.5. Cálculo de las nuevas rutas

Una vez que un enrutador a acumulado un grupo completo de paquetes de estado de enlace, puede construir el grafo de la subred completa porque todos los enlaces están representados. De hecho, cada enlace se representa dos veces, una para cada dirección. Los dos valores pueden promediarse o usarse por separado.

Ahora puede usarse localmente el algoritmo de Dijkstra para construir la trayectoria más corta posible a todos los destinos. Los resultados de este algoritmo pueden instalarse en las tablas de enrutamiento, y reiniciarse la operación normal.

Para una red con *n* enrutadores, cada uno de los cuales tiene *k* vecinos, la memoria requerida para almacenar los datos de entrada es proporcional a *kn*. En las subredes grandes este puede ser un problema. También puede serlo el tiempo de cómputo. Sin embargo, en muchas situaciones prácticas, el enrutamiento por estado de enlaces funciona bien.

Sin embargo, problemas con el hardware o el software pueden causar estragos con este algoritmo. Por ejemplo, si un enrutador indica tener una línea que no tiene, u olvida una línea que si tiene, el grafo de la subred será incorrecto. Si un enrutador deja de reenviar paquetes, o los corrompe al hacerlo, surgirán problemas. Por último, si al enrutador se le acaba la memoria o se ejecuta mal el algoritmo de cálculo de enrutamiento, ocurrirán cosas malas. A medida que la subred crece al orden de decenas o cientos de miles de nodos, la probabilidad de falla ocasional de un enrutador deja de ser insignificante. Lo importante es tratar de limitar el daño cuando ocurra lo inevitable.

El enrutamiento por estado de enlace se usa ampliamente en las redes actuales, por lo que es necesario hablar acerca de los protocolos (que se verá más adelante) que usan este algoritmo. El protocolo OSPF, que es utilizado frecuentemente en Internet utiliza este algoritmo. También lo hace el IS-IS (Intermediate System-Intermediate System. Sistema intermedio. Sistema intermedio).

Eliminado: 89¶

Eliminado: *1*

Eliminado: *5*

2.4. Encaminamiento Jerárquico

A medida que crecen en tamaño las redes, crecen proporcionalmente las tablas de enrutamiento del enrutador. Las tablas que siempre crecen no solo consumen memoria del enrutador, sino que también se necesita más tiempo de CPU para examinarlas, y más ancho de banda para enviar informes de estado entre enrutadores. En cierto momento, la red puede crecer hasta el punto en que ya no es factible que cada enrutador tenga una entrada para cada uno de los demás enrutadores, por lo que el enrutamiento tendrá que hacerse jerárquicamente, como ocurre en la red telefónica.

Al usarse el enrutamiento jerárquico, los enrutadores se dividen en lo que llamaremos **regiones**, donde cada enrutador conoce todos los detalles de la manera de enrutar paquetes a destinos dentro de su propia región, pero no sabe nada de la estructura interna de las otras regiones. Al interconectar 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.

En las redes enormes puede ser insuficiente una jerarquía de dos niveles; puede ser necesario agrupar las regiones en cúmulos, los cúmulos en zonas, las zonas en grupos, etc., hasta que se nos agoten los nombres para los agregados. Como ejemplo de jerarquía multinivel, considere una posible forma de enrutar un paquete de Paute, Azuay a Madrid, España. El enrutador de paute conocería la topología detallada del Azuay, pero podría enviar todo el tráfico exterior al enrutador de Cuenca. El enrutador de Cuenca podría enrutar el tráfico a otros enrutadores del país, pero enviaría el tráfico internacional a Quito. El enrutador de Quito tendría la programación para dirigir todo el tráfico al enrutador del país de destino encargado del manejo de tráfico internacional, digamos en Barcelona. Por último, el paquete encontraría su camino por el árbol de España hasta llegar a Madrid.

Eliminado: 89¶

Eliminado: 1

Eliminado: 5

2.4.1. Tablas de enrutamiento en encaminamiento jerárquico

En la siguiente figura se da un ejemplo cuantitativo de enrutamiento en una jerarquía de dos niveles con cinco regiones. La tabla de enrutamiento completa para el enrutador *1A* tiene 17 entradas, como se muestra en la figura B. Si el enrutamiento es jerárquico, como en la figura C, hay entradas para todos los enrutadores locales, igual que antes, pero las demás regiones se han condensado en un solo enrutador, por lo que todo el tráfico para la región 2 va a través de la línea *1B-2A*, pero el resto del tráfico remoto viaja por la línea *1C-3B*. El enrutamiento jerárquico redujo la tabla de 17 entradas a 7. A medida que crece la razón entre la cantidad de regiones y el número de enrutadores por región, aumentan los ahorros de espacio de la tabla.

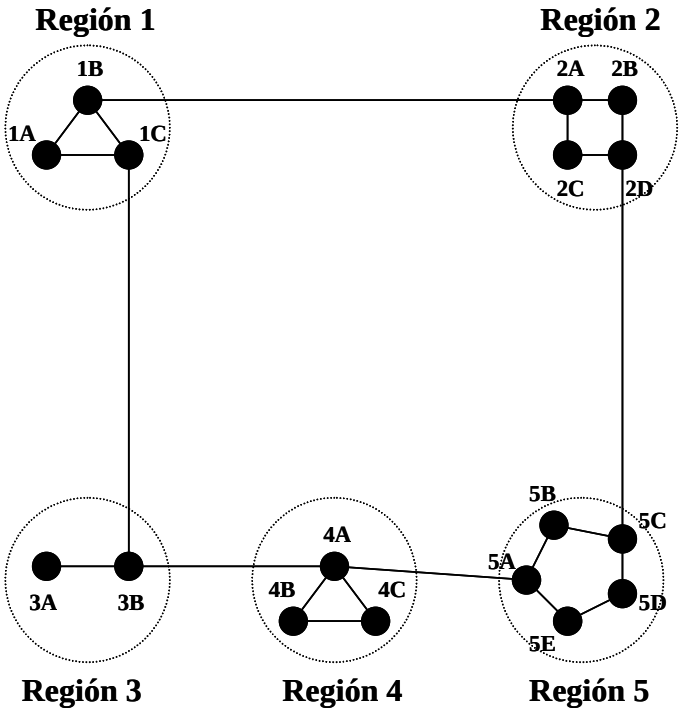


Figura A.

Eliminado: 89¶

Destino	Línea	Escalas
1A	-	-
1B	1B	1
1C	1C	1
2A	1B	2
2B	1B	3
2C	1B	3
2D	1B	4
3A	1C	3
3B	1C	2
4A	1C	3
4B	1C	4
4C	1C	4
5A	1C	4
5B		5
5C	1B	5
5D	1C	6
5E	1C	5

Figura B. Tabla completa para 1A

Destino	Línea	Escalas
1A	-	-
1B	1B	1
1C	1C	1
2	1B	2
3	1C	2
4	1C	3
5	1C	4

Figura C. Tabla jerárquica para 1A

Desafortunadamente, estas ganancias de espacio no son gratuitas, Se paga un precio, y este precio adopta la forma de una longitud de trayectoria mayor. Por ejemplo, la mejor ruta de *1A* a *5C* es a través de la región 2 pero, con el enrutamiento jerárquico, todo el tráfico a la región 5 pasa por la región 3, porque es mejor para la mayoría de los destinos de la región 5.

Al volverse muy grande una sola red, surge una pregunta interesante: ¿cuántos niveles debe tener la jerarquía? Por ejemplo, considere una subred con 720 enrutadores. Si no hay jerarquía, cada enrutador necesita 720 tablas de enrutamiento. Si partimos la subred en 24 regiones de 30 enrutadores, cada enrutador necesitará 30 entradas locales más 23 entradas remotas, para un total de 53 entradas. Si escogemos una jerarquía de 3 niveles, con 8 cúmulos, cada uno de los cuales contiene 9 regiones de 10 enrutadores, cada enrutador necesita 10 entradas para los enrutadores locales, 8 para el enrutamiento a otras regiones dentro de su propio cúmulo y 7 entradas para cúmulos distantes, para un total de 25 entradas. El número óptimo de niveles para una subred de enrutadores es de “ln *N*”, requiriéndose un total de “*e ln N*” entradas por enrutador. También se ha demostrado que el aumento en la longitud media efectiva de trayectoria causada por el enrutamiento jerárquico es lo bastante pequeña como para ser generalmente aceptable.

Eliminado: cada una

Eliminado: entradas

Eliminado: Kamoun y Kleinrock descubrieron que e

Eliminado: n

Eliminado: 89¶

Eliminado: 2

Eliminado: .

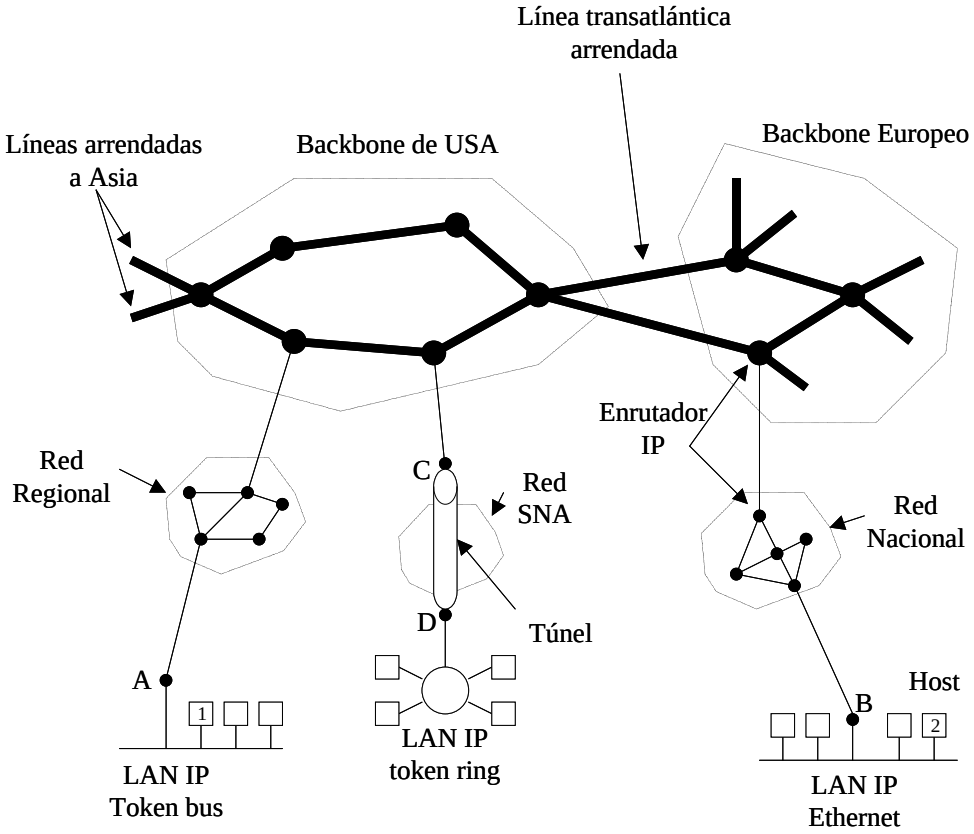
CAPÍTULO 3

Protocolos de Encaminamiento

Con formato: Numeración y viñetas

3.1. *Introducción*

En la capa de red, La Internet puede verse como un conjunto de subredes, o **Sistemas Autónomos** interconectados. No hay una estructura real, pero existen varios backbone principales. Éstos se construyen a partir de líneas de alto ancho de banda y enrutadores rápidos. Conectadas a los backbone hay redes regionales (de nivel medio), y conectadas a estas redes regionales están las LAN de muchas universidades, compañías y proveedores de servicio Internet. En la siguiente figura se presenta un dibujo de esta organización cuasijerárquica.



Con formato

Eliminado: 89¶

El eslabón que mantiene unida la Internet es el protocolo de capa de red IP, (Internet Protocol). A diferencia de la mayoría de los protocolos de capa de red anteriores, éste se diseñó desde el principio con la interconexión de redes en mente. Una buena manera de visualizar la capa de red es la siguiente. Su trabajo es proporcionar un medio de mejor esfuerzo para el transporte de datagramas del origen al destino, sin importar si estas máquinas están en la misma red, o si hay otras redes entre ellas.

La comunicación en Internet funciona como sigue. La capa de transporte toma corrientes de datos y las divide en datagramas. En teoría, los datagramas pueden ser de hasta 64 kbytes cada uno, pero en la práctica por lo general son de unos 1500 bytes. Cada datagrama se transmite a través de Internet, posiblemente fragmentándose en unidades más pequeñas en el camino. Cuando todas las piezas llegan finalmente a las máquinas de destino, son reensambladas por la capa de red, dejando el datagrama original. Este datagrama entonces es entregado a la capa de transporte, que lo introduce en la corriente de entrada del proceso receptor.

El encaminamiento forma parte de la capa de red, pero la función principal de un protocolo de encaminamiento es intercambiar información con otros enrutadores, y en este sentido los protocolos se comportan como si fueran de aplicación.

Eliminado: Es
Eliminado: un

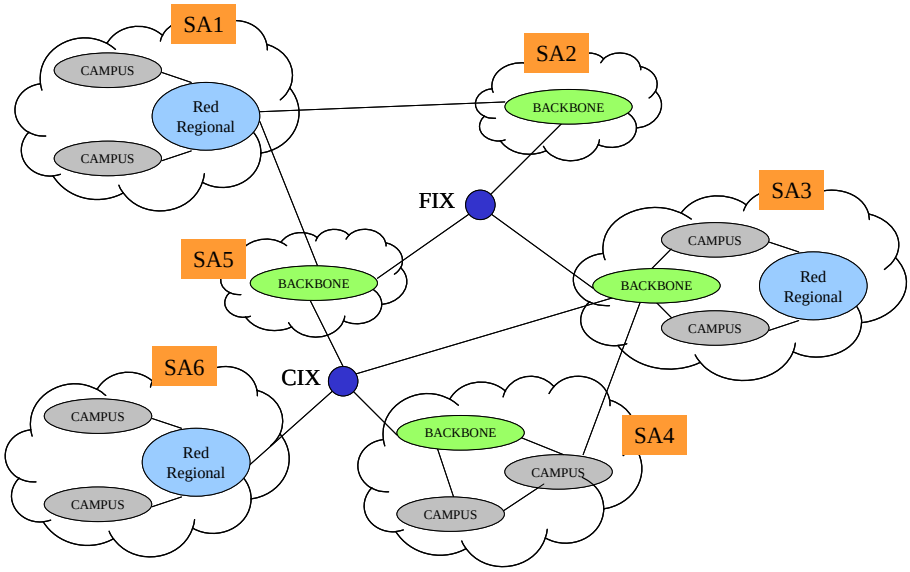
Los protocolos de encaminamiento además son métodos de intercambio de información entre Sistemas Intermedios con el objeto de calcular automáticamente las tablas de encaminamiento de un enrutador.

Entre las principales funciones de un protocolo de encaminamiento tenemos:

- Establecimiento de vecindades: los enrutadores conocen quienes son sus vecinos enviando paquetes “hello”, cuando los otros lo reconocen, intercambian información.
- Distribución y recogida de información: Se envía la información que se conoce de la red, hacia otros enrutadores

- Cálculo de tablas (rutas) mediante un algoritmo de camino más corto (Dijkstra, Bellman-Ford, etc.): Luego de poseer toda la información se establecen las rutas y se crean las tablas de enrutamiento

Como se mencionó anteriormente, el Internet está dividido en Sistemas Autónomos o Dominios de encaminamiento, que son un conjunto de redes gestionadas por una administración común y que comparten una estrategia de encaminamiento común.



CIX (Comercial Interchange) y FIX (Federal Interchange) son Puntos de interconexión (NAP, Network Access Point) en donde se intercambia tráfico de subredes

Internamente en cada uno de estos Sistemas Autónomos se define un tipo de encaminamiento conocido como encaminamiento intradominio o IntraSA. En consecuencia existen protocolos especializados en trabajar dentro de los SA o IGP's (Internal Gateway Protocol). Entre los principales IGP's podemos mencionar: GGP (Gateway to Gateway Protocol), Routing Information Protocol (RIP), Hello (usado en NSFnet), IGRP (Internal Gateway Routing Protocol, propietario de Cisco Systems) y OSPF (Open Short Path First Protocol).

Eliminado: 89¶

De la misma manera, cuando se trata del encaminamiento entre Sistemas Autónomos, se define el encaminamiento interdominio o InterSA en los que se utilizan los protocolos de encaminamiento exterior o EGP (External Gateway Protocol). Entre los principales protocolos interSA tenemos: EGP y BGP (Border Gateway Protocol).

Eliminado: ¶

Eliminado: *2*

3.2. Protocolos de Encaminamiento Interior o Intradominio (IGP's)

Los protocolos de encaminamiento interior se utilizan para intercambiar información de encaminamiento entre enrutadores dentro de un solo Sistema Autónomo . También lo usan los enrutadores que ejecutan protocolos de encaminamiento exterior para recoger información de accesibilidad de la red para el Sistema Autónomo.

Entre los protocolos de encaminamiento interior o intradomino podemos mencionar como los más usados al protocolo Hello, el protocolo RIP y al protocolo OSPF.

Los algoritmos utilizados en los protocolos de enrutamiento interior son el de Vector de distancia, el de Estado de enlace y el de Camino más corto.

Con formato: Numeración y viñetas

3.2.1. Protocolo RIP

Existen dos versiones de RIP. La versión 1 (RIP-1) es un protocolo destacado ampliamente con sus limitaciones. La versión 2 (RIP-2) es una versión mejorada diseñada para aliviar las limitaciones de RIP hasta que sea altamente compatible con él.

El término RIP se usa para referirse a la versión 1, mientras que RIP-2 se refiere a la versión 2. En este documento utilizaremos RIP siempre para referirnos a la versión 1 del protocolo RIP.

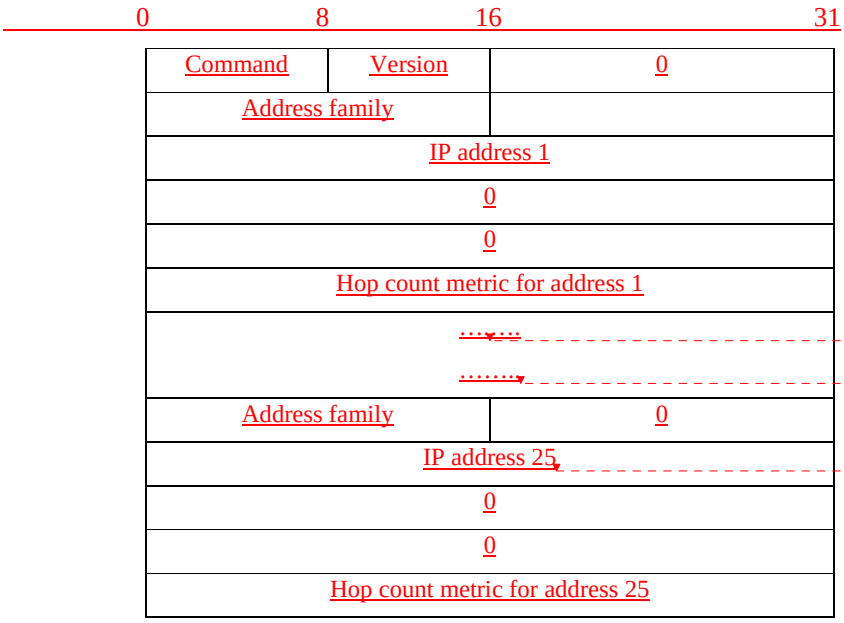
Eliminado: 2

3.2.1.1. Protocolo de Información de Enrutamiento, V. 1 (RIP, RIP-1)

RIP es una implementación directa del algoritmo de enrutamiento vector de distancia, para redes locales. La comunicación RIP usa UDP como protocolo de transporte, con número de puerto 520 como puerto de destino. RIP opera en uno de los dos modos siguientes:

activo (normalmente lo usan los routers) y pasivo (normalmente lo usan los hosts). La diferencia entre los dos se explica más abajo.

Los mensajes RIP se envían en datagramas UDP y cada uno contiene 25 parejas de números.



Se pueden listar entre 1 y 25 rutas en un mensaje RIP. Con 25 rutas el mensaje es de 504 bytes (25x20+4) que es el tamaño máximo del mensaje que puede transmitirse en un datagrama UDP de 512 bytes.

Command: es 1 para una petición RIP o 2 para una respuesta RIP.

Versión: es 1.

Address family: es 2 para direcciones IP.

IP Address: es la dirección IP para esta entrada de enrutamiento: un host o una subred (en cuyo caso el número de host es cero).

Hop count metric: es la número de saltos al destino. El contador de saltos para una interfaz conectada directamente es 1, y cada router intermedio lo incrementa en 1 hasta un máximo de 15, un 16 indica que no existe ruta hacia el destino.

Eliminado: 89¶

Eliminado: ¶

Eliminado: ¶

Eliminado: ¶

Eliminado:

Eliminado: *Orden*

Eliminado: ¶

Con formato

Con formato

Eliminado: *Familia de direcciones*

Con formato

Eliminado: ¶

Eliminado: *dirección*

Con formato

Eliminado: ¶

Con formato

Eliminado: *Métrica de salto*

Con formato

Eliminado: ¶

Eliminado: 89¶

Ambos participantes RIP, activo y pasivo, escuchan todos los mensajes emitidos y actualizan sus tablas de enrutamiento según el algoritmo vector-distancia descrito anteriormente.

Operaciones Básicas

Con formato

- Cuando RIP comienza envía un mensaje para cada uno de sus vecinos (sobre el puerto UDP bien-conocido 520) solicitando una copia de la tabla de enrutamiento del vecino. Este mensaje es una pregunta (orden se pone a 1) con una familia de direcciones de 0 y una métrica de 16. Los enrutadores de los vecinos devuelven una copia de sus tablas de enrutamiento.

Con formato: Numeración y viñetas

Eliminado: routers

- Cuando RIP está en modo activo envía todo o parte de su tabla de enrutamiento a todos los enrutadores de sus vecinos (mediante broadcasting y/o enviándolo sobre cualquier enlace punto-a-punto hacia sus vecinos). Esto se hace cada 30 segundos. La tabla de enrutamiento se envía como respuesta (orden es 2, incluso aunque no se solicite).

Con formato: Numeración y viñetas

Eliminado: routers

- Cuando RIP descubre que una métrica ha cambiado, emite el cambio a otros enrutadores.

Con formato: Numeración y viñetas

Eliminado: routers

- Cuando RIP recibe una respuesta, el mensaje se valida y la tabla de enrutamiento local se actualiza si es necesario.

Con formato: Numeración y viñetas

  - Para mejorar el rendimiento y la confiabilidad, RIP especifica que una vez que un enrutador (o un host) aprende la ruta de otro enrutador, debe mantener esa ruta hasta que aprenda la mejor (con un coste estrictamente bajo). Esto impide aquellas rutas oscilatorias entre dos o más caminos de igual coste.

Con formato: Numeración y viñetas

Eliminado: router

Eliminado: router

- Cuando RIP recibe una petición, distinta de la solicitud de su tabla, se devuelve como respuesta la métrica para cada entrada de dicha petición fijada al valor de la tabla local de encaminamiento. Si no existe ruta en la tabla local, se pone a 16.

- Las rutas que RIP aprende de otros enrutadores expiran a menos que se vuelvan a difundir en 180 segundos (6 ciclos de broadcast). Cuando una ruta expira, su métrica se pone a infinito, la invalidación de la ruta se difunde a los vecinos, y 60 segundos más tarde, se borra de la tabla.

Limitaciones

RIP no está diseñado para resolver cualquier posible problema de encaminamiento. Estas limitaciones técnicas se describen como graves y se están analizando posibles protocolos que podrían solucionar estos problemas. Dichos protocolos podrían ser OSPF e IS-IS. Sin embargo, RIP está muy extendido y es probable que permanezca sin sustituir durante algún tiempo. Tiene las siguientes limitaciones:

- El coste máximo permitido en RIP es 16, que signifca que la red es inalcanzable. De esta forma, RIP es inadecuado para redes grandes (es decir, aquellas en las que la cuenta de saltos puede aproximarse perfectamente a 16).

- RIP no soporta máscaras de subred de longitud variable. En un mensaje RIP no hay ningún modo de especificar una máscara de subred asociada a una dirección IP.

- RIP carece de servicios para garantizar que las actualizaciones proceden de enrutadores autorizados. Es un protocolo inseguro.

- RIP solo usa métricas fijas para comparar rutas alternativas. No es apropiado para situaciones en las que las rutas necesitan elegirse basándose en parámetros de tiempo real tales como el retardo, la fiabilidad o la carga.

Eliminado: 89¶

Con formato: Numeración y viñetas

Eliminado: otro than one for the entire table, it is returned as the response with the metric for each entry set to the value from the local routing table. If no route exists in the local table, the metric is set to 16.

Eliminado: RIP routes learned from other routers time out unless they are re-advertised within 180 seconds (6 broadcast cycles). When a route times out, its metric is set to infinity, the invalidation of the route is broadcast to the router's neighbors, and 60 seconds later, the route is deleted from the local routing table.

Con formato: Numeración y viñetas

Con formato

Eliminado: RIP is not designed to solve every possible routing problem.RFC 1720 (STD 1) describes these technical limitations of RIP as ``serious'' and the IETF es evaluating candidates for a new standard ``open'' protocol to replace RIP. Possible candidates include OSPF (see Open Shortest Path First Protocol (OSPF) Version 2) and OSI IS-IS (see OSI Intermediate System to Intermediate System (IS-IS)). However, RIP is widely deployed and therefore is unlikely to be completely replaced for some time. RIP has the following specific limitations:

Con formato: Numeración y viñetas

Con formato: Numeración y viñetas

Con formato: Numeración y viñetas

Con formato: Numeración y viñetas

Eliminado: 89¶

Con formato: Numeración y viñetas

- El protocolo depende de la *cuenta al infinito* para resolver algunas situaciones iniciales. Como se describió anteriormente (algoritmo de vector de distancia), la resolución de un bucle requeriría mucho tiempo (si la frecuencia de actualización fuese limitada) o mucho ancho de banda (si las actualizaciones se enviasen por cada cambio producido). A medida que crece el tamaño del dominio, la inestabilidad del algoritmo *vector de distancia* de cara al cambio de topología se hace patente. RIP especifica mecanismos para minimizar los problemas con la *cuenta hasta infinito* que permiten usarlos con dominios mayores, pero eventualmente su operatividad será nula. No existe un límite superior prefijado, pero a nivel práctico este depende de la frecuencia de cambios en la topología, los detalles de la topología de la red, y lo que se considere como un intervalo máximo de tiempo para que la topología de encaminamiento se estabilice.

Con formato

3.2.1.2. Protocolo de Información de Enrutamiento, V. 2 (RIP-2)

RIP-2 es menos potentes que otros IGP's recientes tales como OSPF, pero tiene las ventajas de una fácil implementación y menores factores de carga.

La intención de RIP-2 es proporcionar una sustitución directa de RIP que se pueda usar en redes pequeñas y medianas, en presencia de subnetting variable o supernetting y, sobretodo, que pueda interoperar con RIP-1.

RIP-2 aprovecha que la mitad de los bytes de un mensaje RIP están reservados (deben ser cero) y que la especificación original estaba diseñada con las mejoras en la mente de los desarrolladores, particularmente en el uso del campo de versión.

Un área notable en la que este no es el caso es la interpretación del campo de métrica. RIP-1 lo especifica con un valor de 0 a 16 almacenado en un campo de 4 bytes. Por compatibilidad, RIP-2 preserva esta definición, lo que significa que interpreta 16 como infinito, y desperdicia la mayor parte del rango de este campo.

El formato del mensaje RIP-2 es el siguiente:

0 8 16 31

Command	Version	0
X'FFFF'		Authentic Type
Authentication data (16 bytes)		
Address family		Route tag 1
IP address 1		
Subnet mask 1		
Next hop 1		
Hop count metric for address 1		
……. …….		
Address family		Route tag 24
IP address 24		
Subnet mask 24		
Next hop 24		
Hop count metric for address 24		

La primera entrada del mensaje puede ser una entrada de autentificación, como se muestra aquí, o una ruta como en el mensaje RIP. Si la primera entrada es de autentificación, sólo se puede incluir 24 rutas en el mensaje; de otro modo, el máximo es de 25, como en RIP.

Los campos del mensaje RIP-2 son los mismos que en RIP excepto los siguientes:

Version: es 2. Le indica RIP-1 al enrutador que ignore los campos reservados, los que deben ser cero (si el valor es 1, los enrutadores deben desechar los mensajes con valores distintos de cero en estos campos, ya que los originó un enrutador que dice ser RIP, pero envía mensajes que no cumplen el protocolo).
Address family: Puede ser X'FFFF' solo en la primera entrada, inidcando que se trata de una entrada de autentificación.

Eliminado: 89¶

Authentication type: Define como se han de usar los restantes 16 bytes. Los únicos tipos definidos son 0, indicando ninguna autentificación, y 2 indicando que el campo contiene datos de password.

Authentication data: El password es de 16 bytes, texto ASCII plano, alineado a la izquierda y rellenado con caracteres nulos ASCII (X'00').

Rote tag: Es un campo dirigido a la comunicación de información acerca del origen de la información de encaminamiento. Está diseñado para la interoperabilidad entre RIP y otros protocolos de encaminamiento. Las implementaciones de RIP-2 deben conservarlo, aunque RIP-2 no especifica cómo se debe usar.

Subnet mask: La máscara de subred asociada con la subred a la que se refiere esta entrada.

Next Hop: Una recomendación acerca del siguiente salto que el enrutador debería usar para enviar datagramas a la subred o al host dado en la entrada.

Para asegurar una interoperabilidad segura con RIP, existen las siguientes restricciones para los enrutadores RIP-2 que transmiten sobre una interfaz de red en la que un enrutador RIP puede escuchar y operar con mensajes RIP.

1. La información interna a una red nunca se debe anunciar a otra red.
2. La información acerca de una subred más específica no se debe anunciar donde los enrutadores vean una ruta de host.
3. Las rutas a superredes (rutas con una máscara de subred más corta que la máscara natural de la red) no se deben anunciar en los sitios en los que puedan ser malentendidas por los enrutadores RIP.

Con formato: Numeración y viñetas
Con formato
Con formato: Numeración y viñetas
Con formato
Con formato: Numeración y viñetas
Con formato

RIP-2 soporta además el multicast con preferencia al broadcast. Esto puede reducir la carga de los hosts que no están a la escucha de mensajes RIP-2. Esta opción es configurable para cada interfaz para asegurar un uso óptimo de los servicios RIP-2 cuando un enrutador conecta redes mixtas RIP-1/RIP-2 CON REDES rip-2. Similarmente, el uso de la autentificación en entornos mixtos se puede configurar para adecuarse a los requerimientos locales.

Eliminado: 89¶

Nota: Ni RIP-1 ni RIP-2 son adecuados para ser usados como IGP's en un SA en el que el valor de 16 sea demasiado bajo para ser considerado infinito, ya que los valores altos del infinito exacerban el problema de la cuenta hasta infinito.

El algoritmo de estado de enlace, más sofisticado, usado en OSPF y en IS-IS proporciona una solución de encaminamiento mucho mejor cuando el SA es lo bastante largo para tener una cuenta de saltos cercana a 16

3.2.2. Protocolo OSPF

Como se mencionó antes, La Internet se compone de una gran cantidad de sistemas autónomos (SA). Cada SA es operado por una organización diferente y puede usar internamente su propio algoritmo de enrutamiento. Por ejemplo, las redes internas de tres empresas diferentes generalmente se verían como tres Sistemas Autónomos si las tres estuvieran en Internet. Las tres puede usar algoritmos de enrutamiento diferentes internamente. No obstante, la existencia de estándares, aún para enrutadores internos, simplifica la implementación en las líneas divisorias entre los SA's y permite la reutilización de código. Ahora revisaremos el enrutamiento dentro de un SA, luego, entre varios SA's. Al algoritmo utilizado dentro un SA se le denomina *protocolo de pasarela interior* y al algortimo entre diferentes SA's *protocolo de pasarela exterior.*

El protocolo de pasarela interior original de Internet fue un protocolo de vector de distancia (RIP) basado en el algoritmo Bellman-Ford. Este protocolo funcionó bien en sistemas pequeños, pero no muy bien a medida que los SA se volvieron más grandes; también padecía el problema de conteo a infinito y generalmente de una convergencia lenta por lo que fue reemplazado por un algoritmo de estado de enlace. Luego el sucesor de este aparece en 1988 y se lo denomina **OSPF** (Open Shortest Path First, *abrir primero la trayectoria más corta*), se convirtió en estándar en 1990. Ahora es utilizado en muchos enrutadores y es el principal protocolo de pasarela interior.

En el diseño de este protocolo habían que cumplir con una serie de requisitos. Primero, el algoritmo tenía que publicarse como literatura abierta, de ahí la "O" (de Open) en OSPF. No serviría una solución patentada, que fuera propiedad de una compañía. Segundo, el

Eliminado: ¶
The maximum cost allowed in RIP is 16 which means that the network is unreachable. Thus RIP is inadequate for large networks (that is, those in which legitimate hop counts approach 16). ¶
RIP does not support variable length subnet masks (variable subnetting). There is no facility in a RIP message to specify a subnet mask associated with the IP address. ¶
RIP has no facilities to ensure that routing table updates come from authorized routers. It is an unsecure protocol. ¶
RIP only uses fixed metrics to compare alternative routes. It is not appropriate for situations where routes need to be chosen based on real-time parameters such as measured delay, reliability, or load. ¶
The protocol depends upon counting to infinity to resolve certain unusual situations. As described earlier (Vector-Distance), the resolution of a loop would require either much time (if the frequency of updates was limited) or much bandwidth (if updates were sent whenever changes were detected). As the size of the routing domain grows, the instability of the vector-distance algorithm in the face of changing topology become ... [2]

Con formato
Con formato
Con formato
Con formato
Con formato
Con formato
Con formato
Con formato
Con formato
Con formato
Con formato
Con formato
Con formato
Con formato
Con formato
Con formato
Con formato
Con formato
Con formato

Eliminado: 2

Eliminado: 89¶

protocolo tenía que reconocer una variedad de métricas de distancia, incluidas distancia física, retardo y otras. Tercero, tenía que ser un algoritmo dinámico, uno que se adoptara a los cambios de topología, rápida y automáticamente.

Cuarto, tenía que reconocer el enrutamiento basado en el tipo de servicio. Tenía que ser capaz de enrutar el tráfico de tiempo real de una manera y otros tipos de tráficod e otra manera. El protocolo IP tiene el campo *tipo de servicio*, pero ningún protocolo de enrutamiento existente lo usaba.

Quinto, y relacionado con lo anterior, este protocolo tenía que efectuar equilibrio de cargas, dividiendo la carga entre varias líneas. La mayoría de los protocolos previos enviaban todos los paquetes a través de la mejor ruta. La segunda mejor ruta no se usaba en lo absoluto. En muchos casos, la división de la carga a través de varias líneas produce un mejor desempeño.

Sexto, se requería el reconocimiento de sistemas jerárquicos. En 1988, la Internet se había hecho tan grande que no se podía esperar que un solo enrutador conociera la topología completa. Este protocolo tenía que diseñarse de modo que ningún enrutador tuviera que conocerla completamente.

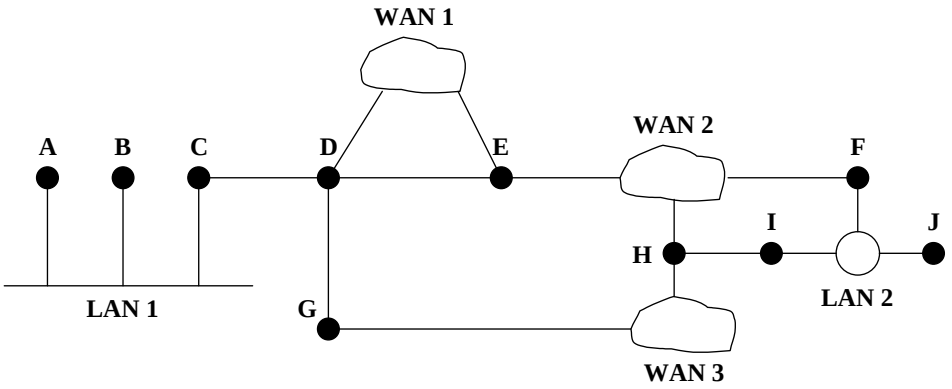
Séptimo, se requería un mínimo de seguridad para evitar que los piratas de red burlaran a los enrutadores enviándoles información de enrutamiento falsa. Por último, se requería un mecanismo para manejar los enrutadores que se conectaran a Internet a través de un "túnel" (ver Anexo 2).

Eliminado: 4

El OSPF reconoce tres tipos de conexiones y redes:

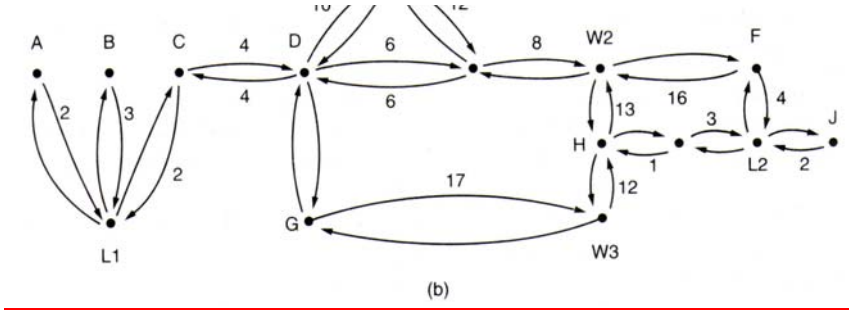
1. Líneas punto a punto entre dos enrutadores (exactamente).
2. Redes multiacceso con difusión (por ejemplo, la mayoría de las LAN).
3. Redes multiacceso sin difusión (por ejemplo, la mayoría de las WAN de conmutación de paquetes).

Una red **multiacceso** es una que puede tener varios enrutadores, cada uno de los cuales puede comunicarse directamente con todos los demás. Todas las LAN y WAN tienen esta propiedad. En la siguiente figura se muestra un SA que contiene los tres tipos de redes. Nótese que los *hosts* generalmente no desempeñan ningún papel en el OSPF.



Sistema Autónomo

El OSPF funciona haciendo una abstracción del conjunto de redes, enrutadores y líneas en un grafo dirigido en el que a cada arco se le asigna un costo (distancia, retardo, etc.). Entonces se calcula la trayectoria más corta con base en los pesos de los arcos. Una conexión en serie entre dos enrutadores se representa mediante un par de arcos, uno en cada dirección. Sus pesos pueden ser diferentes. Una red multiacceso se representa mediante un nodo para la red misma más un nodo para cada enrutador. Los arcos del nodo de red a los enrutadores tienen un peso de 0 y se omiten del grafo. En la figura a continuación se muestra la representación gráfica de la red de la figura anterior. Lo que hace fundamentalmente el OSPF es representar la red como un grafo de este tipo y luego calcular la trayectoria más corta de un enrutador a todos los demás.



Representación con grafos

Muchas de los SA del Internet son grandes y nada fácil de manejar. El OSPF permite su división en **áreas** numeradas, donde un área es una red o un grupo de redes contiguas. Las áreas no se traslapan, pero no necesitan ser exhaustivas, es decir, algunos enrutadores podrían no pertenecer a ningún área. Un área es una generalización de una subred. Fuera de un área, su topología y detalles no son visibles.

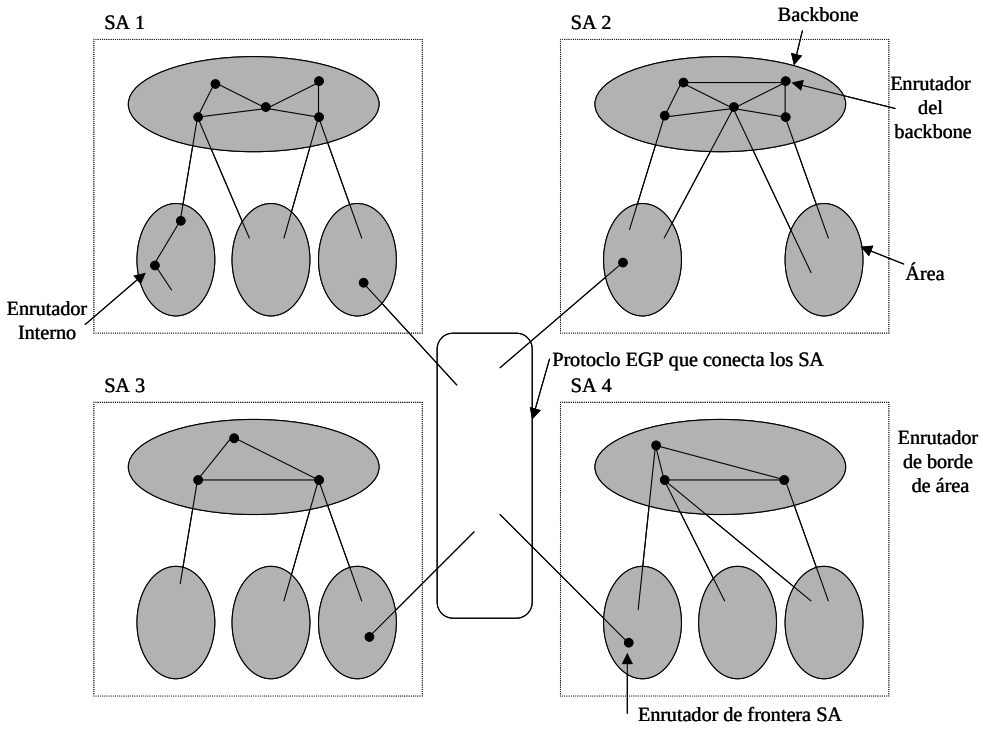
Cada SA tiene un área de **backbone**, llamada área 0. Todas las áreas se conectan al *backbone*, posiblemente mediante túneles, por lo que hay la posibilidad de ir de cualquier área del SA a cualquier otra a través del backbone. Un túnel se representa en el grafo como un arco y tiene un costo. Cada enrutador conectado a dos o más áreas es parte del backbone. Al igual que en otras áreas, la topología del backbone no es visible desde fuera del backbone.

Dentro de un área, cada enrutador tiene la misma base de datos de estado de enlace y ejecuta el mismo algoritmo de trayectoria más corta; su tarea principal es calcular la trayectoria más corta de sí mismo a todos los enrutadores del área, incluido el enrutador que está conectado al backbone, de los cuales debe haber cuando menos uno. Un enrutador que se conecta a dos áreas necesita las bases de datos de ambas áreas y debe ejecutar para cada una por separado el algoritmo de trayectoria más corta.

La manera en que el OSPF maneja el enrutamiento de tipo de servicio es teniendo varios grafos, uno etiquetado con los costos cuando la métrica es el retardo, otro etiquetado con los costos cuando la métrica es el rendimiento, y uno más etiquetado con los costos cuando la métrica es la confiabilidad. Aunque esto triplica el cálculo necesario, permite rutas separadas para optimar el retardo, el rendimiento y la confiabilidad.

Durante la operación normal, pueden necesitarse tres tipos de rutas: intraárea, interárea e interSA. Las rutas intraárea son las más fáciles, dado que el enrutador de origen ya conoce la trayectoria más corta al enrutador de destino. El enrutamiento interárea siempre procede en tres pasos: va del origen al backbone, pasa a través del backbone al área de destino y va al destino. Este algoritmo obliga a una configuración en estrella en el OSPF, siendo el

backbone el centro y las demás áreas los rayos. Los paquetes se enrutan del origen al destino “como vienen”. No se encapsulan ni se envían por túnel, a menos que vayan a un área cuya única conexión al backbone sea un túnel. En la siguiente figura se muestra parte de la Internet con SA y áreas.



Relación entre los SA, los backbone y las áreas en el OSPF

El OSPF distingue cuatro clases de enrutadores:

1. Enrutadores internos que están contenidos en una sola área.
2. Enrutadores de borde de área que conectan dos o más áreas
3. Enrutadores de backbone que están en el backbone
4. Enrutadores de frontera de área que hablan con los enrutadores de otros áreas.

Se permite que estas clases se traslapen. Por ejemplo, todos los enrutadores de borde son automáticamente para el backbone. Además, un enrutador que está en el backbone pero no

Eliminado: 89¶

es parte de ningún otro área también es un enrutador interno. En la figura se ilustran ejemplos de las cuatro clases de enrutadores.

Eliminado: ¶

Al arrancar un enrutador, envía mensajes de HOLA por todas sus líneas punto apunto y los multitransmite por las LAN al grupo que consiste en todos los demás enrutadores. En las WAN, el enrutador requiere cierta información de configuración para saber con quién tiene que hacer contacto.A partir de las respuestas, cada enrutador aprende quiénes son sus vecinos.

El OSPF funciona intercambiando información entre enrutadores adyacentes, que no es lo mismo que entre enrutadores vecinos. En particular, es ineficiente hacer que todos los enrutadores de una LAN hablen con todos los enrutadores de otra LAN. Para evitar esta situación, se elige un enrutador como *enrutador designado*, el cual se dice que es adyacente a todos los demás enrutadores, e intercambia información con ellos. Los enrutadores vecinos que no son adyacentes no intercambian información entre ellos. Siempre se mantiene actualizado un enrutador designado de respaldo para facilitar la transición en caso de que el enrutador designado primario se caiga.

Durante la operación normal, cada enrutador inunda periódicamente con mensajes de ACTUALIZACIÓN DE ESTADO DE ENLACE (LINK STATE UPDATE) a todos sus enrutadores adyacentes. Este mensaje indica el estado del enrutador y proporciona los costos usados en la base de datos topológica.. Los mensajes de inundación se reconocen a fin de hacerlos confiables. Cada mensaje tiene un número de secuencia, por lo que un enrutador puede ver si un mensaje de ACTUALIZACIÓN DE ESTADO DE ENLACE de entrada es más viejo o más nuevo que el que tiene actualmente. Los enrutadores también pueden enviar estos mensajes cuando se activa o se desactiva una línea y cuando cambian sus costos.

Los mensajes de DESCRIPCIÓN DE LA BASE DE DATOS (DATABASE DESCRIPTION) dan los números de secuencia de todas las entidades de estado de enlace guardadas actualmente por el transmisor. Al comparar su propios valores con los del

Eliminado: 89¶

transmisor, el receptor puede determinar quién tiene los valores más recientes. Estos mensajes se usan al activar la línea.

Cualquiera de las partes puede solicitar información de estado de enlace de la otra mediante mensajes de SOLICITUD DE ESTADO DE ENLACE (LINK STATE REQUEST). El resultado neto de este algoritmo es que cada par de enrutadores adyacentes verifica quien tiene los datos más recientes, y de esta manera se distribuye información nueva a través del área. Todos estos mensajes se envían como paquetes IP en bruto. Los cinco tipos de mensaje se resumen a continuación.

Tipo de Mensaje	Descripción
Hola (Hello)	Sirve para descubrir quiénes son los vecinos
Actualización de estado de enlace	Proporciona los costos del transmisor a sus vecinos
Reconocimiento de estado de enlace	Reconocimiento de la actualización de estado de enlace
Descripción de la base de datos	Anuncia las actualizaciones que tiene el transmisor
Solicitud de estado de enlace	Solicita información del compañero

Por último, podemos juntar todas las piezas. Mediante inundación, cada enrutador informa a los demás enrutadores de su área acerca de sus vecinos y sus costos. Esta información permite a cada enrutador construir el grafo de su(s) área(s) y calcular la trayectoria más corta. El área de backbone también hace esto. Además, los enrutadores de backbone aceptan información de los enrutadores de borde de área a fin de calcular la mejor ruta de cada enrutador de backbone a los demás enrutadores. Esta información se propaga de regreso a los enrutadores de borde de área, quienes la divulgan en sus áreas. Usando esta información, un enrutador a punto de enviar un paquete interárea puede seleccionar el mejor enrutador de salida al backbone.

Eliminado: ¶
¶
Eliminado: 2

3.2.3. Protocolo IS-IS (Intermediate System - Intermediate System)

IS-IS es un protocolo similar a OSPF, también emplea el algoritmo de estado de enlace, primero el camino más corto. Sin embargo IS-IS es un protocolo OSI usado para los paquetes CLNP (Connectionless Network Protocol) en un dominio de encaminamiento. CLNP es el protocolo OSI más comparable a IP.

Eliminado: 89¶

El IS-IS integrado extiende IS-IS para compararse a TCP/IP. Su meta es proporcionar un solo (y eficiente) protocolo de encaminamiento para TCP/IP y para OSI. Su diseño hace uso del protocolo de encaminamiento OSI IS-IS, aumentando con información IP específica, y proporciona apoyo explícito para el subnetting IP, máscaras de red variables y encaminamiento externo, además de recurso para la autentificación. El IS-IS integrado se basa en el mismo algoritmo de encaminamiento que OSPF.

No emplea encapsulación mutua de los paquetes IP y CLNP: ambos tipos se envían tal como son, ni cambia el comportamiento del enrutador como ambas pilas de protocolos podrían esperar. Se comporta como un IGP en una red TCP/IP y en una red OSI. El único cambio es la adción de información adicional relacionada con IP.

IS-IS agrupa las redes en dominios de modo análogo a OSPF. Un dominio de encaminamiento (o Sistema Autónomo) se subdivide en áreas, igual que en OSPF.

Algunas características de IS-IS son:

- Los enrutadores se dividen en enrutadores de nivel 1, que no saben nada de la topología fuera de sus áreas, y de nivel 2, que conocen la topología de nivel superior, pero no saben nada de la topología de dentro de las áreas, a menos que sean también enrutadores de nivel 1.

Con formato: Numeración y viñetas

- Un enrutador de nivel 1 puede pertenecer a más de un área, pero a diferencia de OSPF esto no se hace con propósitos de encaminamiento sino para facilitar la gestión del dominio, y normalmente por poco tiempo. Un enrutador de nivel 1 reconoce a otro como un vecino si están en la misma área.

Con formato: Numeración y viñetas

- Un enrutador de nivel 2 reconoce a todos los demás enrutadores de nivel 2 como vecinos. Un enrutador de nivel 2 puede ser también un enrutador de nivel en un área, pero no en más.

Con formato: Numeración y viñetas

Eliminado: 89¶

- Un enrutador de nivel 1 en IS-IS no puede tener un enlace con un enrutador externo.

Con formato: Numeración y viñetas

- Hay una troncal de nivel 2 que contiene todos los enrutadores de nivel 2, pero a diferencia de OSPF, debe estar conectada físicamente.

Con formato: Numeración y viñetas

- El esquema de dirección OSI identifica explícitamente el área objetivo de un paquete, permitiendo una selección sencilla de las rutas del modo siguiente:
  - Los enrutadores de nivel 2 encaminan hacia el área sin importarles su estructura interna.
  - Los enrutadores de nivel 1 encaminan hacia el destino si está en su área, o al enrutador de nivel 2 más cercano si no es así.

Con formato: Numeración y viñetas

IS-IS integrado permite una mezcla considerable de las dos pilas de protocolo, sujeto a ciertas restricciones sobre la topología. Se definen tres tipos de rutas:

IP-only: Un enrutador que usa IS-IS como protocolo de encaminamiento y para IP y no soporta protocolos OSI (por ejemplo, tales enrutadores no serían capaces de transmitir paquetes CLNP).

OSI-only: UN enrutador que usa IS-IS como protocolo de encaminamiento para OSI pero no usa IP.

Dual: Un enrutador que usa IS-IS como un único protocolo de encaminamiento integrado tanto para IP como para OSI.

Es posible tener un dominio mixto que contenga enrutadores IS-IS, algunos de los cuales son *IP-only*, algunos *OSI-only*, y algunos del tipo *dual*. Cada área dentro de un dominio se configura como *OSI*, *IP* o *dual*. Las áreas que han de soportar tráfico mixto deben tener todos los enrutadores de nivel 1 del tipo *dual*. Similarmente, los enrutadores de nivel 2 en un dominio mixto deben ser dual si el tráfico mixto se tiene que encaminar entre áreas.

Con formato

Con formato

Con formato

Con formato: Numeración y viñetas

Eliminado: 89¶

Eliminado: ¶
Encaminamiento Integrado¶

3.2.4. Protocolo Hello

La comunicación en el protocolo Hello se hace por mensajes Hello sobre datagramas IP. El número de protocolo de Hello es el 63 (reservado para cualquier red local).

El protocolo Hello es significativo parcialmente debido a su amplia distribución durante la expansión de Internet y parcialmente porque es un ejemplo de algoritmo vector de distancia que no usa cuenta de saltos como RIP sino retardos de red como métrica.

Un host físico DCN (Distributed Computer Network) es un procesador compatible con el PDP11 que soporta un número de procesos cooperativos secuenciales, a cada uno de los cuales se le da un Id unívoco de 8 bits llamado su ID de puerto. Cada host DCN contiene uno o más procesos de Internet, cada uno de los cuales soporta un host virtual dado un ID de 8 bits, llamado el ID de host. Existe una correspondencia uno a uno entre las direcciones de Internet y los ID de hosts. Todos los hosts físicos DCN se identifican por su ID de host con el fin de detectar bucles al actualizar tablas, que establecen los caminos de mínimo retardo entre los hosts virtuales.

Cada host físico tiene dos tablas:

Tabla de Host.- Contiene estimaciones del retardo de viaje de ida y vuelta y un desplazamiento lógico de reloj (es decir, la diferencia entre el reloj lógico de este host y el del emisor). Se indexa por el número de hosts. Se mantiene dinámicamente actualizaciones generadas por mensajes Hello periódicos (de 1 a 30 segundos).

Tabla de red.- Contiene una entrada para cada red vecina conectable a la red local y a otras redes concretas que no sean vecinas. Cada entrada contiene el número de red, además del número de host del enrutador (localizado en la red local) para esa red. Esta tabla se inicializa en tiempo de configuración para todos los hosts excepto en aquellos que soporten los protocolos de encaminamiento GGP o EGP. En estos casos, se actualiza como parte de la operación de encaminamiento.

Eliminado: 89¶

Eliminado: ¶

Además, las entradas de ambas tablas las pueden cambiar los comandos del operador. El retardo y el desplazamiento estimados son actualizados por mensajes Hello intercambiados en los enlaces que conectan los vecinos físicos.

A continuación se muestra el formato de un mensaje Hello:

0	16	24	31
Checksum		Date	
Time			
Timestamp		L Offset	# hosts
Delay 1		Offset 1	
…		…	
…		…	
Delay n		Offset n	

Eliminado:

En donde:

Checksum: contiene un checksum cubriendo los campos indicados.

Date: es la fecha local del host.

Time: es la hora local del host.

Timestamp: usado en cálculos del tiempo de viaje.

L Offset: contiene el desplazamiento del bloque de entradas de direcciones de Internet usado en la red local.

Con formato

 # Hosts: contiene el número de entradas de la tabla de host siguiente.

Delay n: retardo hasta el host n.

Offset n: offset para el host n (diferencia entre los relojes).

Consideremos ahora los dos pasos principales del protocolo Hello.

3.2.4.1. Cálculo del retardo de viaje

Periódicamente cada host envía un mensaje Hello a su vecino en cada enlace común. Para cada uno de estos enlaces el emisor guarda un conjunto de variables de estado, incluyendo

Eliminado: 89¶

una copia del campo dirección fuente del último mensaje Hello recibido. Al construir un mensaje Hello el emisor fija el campo de destino a su variable de estado y el de dirección fuente su propia dirección. Luego rellena los campos fecha y ahora a partir de su reloj y el sello de tiempo de otra variable de estado. Finalmente copia el retardo y los valores de offset de su tabla de host al mensaje.

Los cálculos de retardo del viaje se realizan cuando el host recibe el mensaje. Cada enlace tiene asignado una variable interna de estado, que se actualiza a la recepción de cada mensaje Hello; esta variable toma el valor del campo hora, menos la hora actual de ese momento. Cuando se transmite el siguiente mensaje Hello, el valor asignado al campo sello de tiempo se computa como los 16 bits de orden inferior de esta vriable menos la hora actual. El retardo se calcula como los 16 bits de orden inferior de la hora actual menos el valor de sello de tiempo.

Con formato

3.2.4.2. Actualizaciones del host

Cuando llega un mensaje Hello, lo que da lugar al cálculo de un retardo de viaje, se efectúa un proceso de actualización del host. Consiste en añadir el retardo a cada una de las *n* entradas de retardos en el mensaje Hello y en comparar cada uno de esos valores con el campo retardo (delay) de la tabla de host correspondiente. Cada entrada se actualiza según las siguientes reglas:

- Si el enlace conecta con otro host en la misma red y el ID de puerto del proceso de salida del enlace coincide con el campo ID del puerto de entrada, se actualiza la entrada.
- Si el enlace conecta a otro host en la misma red y el ID de puerto del proceso de salida del enlace no coincide con el número de puerto de la entrada y el retardo calculado es menor que el campo de retardo del host de la tabla de host en al menos un umbral de conmutación especificado (habitualmente 100 ms), se actualiza la entrada. Por ejemplo, si el host A envía a B un mensaje Hello, y el retardo actual de B para alcanzar es mayor que el retardo de A a D más el retardo de B a A, B cambia su ruta y envía el tráfico a D por A.

Con formato: Numeración y viñetas

Con formato

Con formato

Eliminado: 89¶

El propósito del umbral de conmutación es evitar (además de ser una especificación del retardo mínimo) conmutaciones innecesarias entre enlaces y bucles transitorios que pueden ocurrir debido a variaciones normales en los retardos de propagación.

Con formato

3.2.5. Encaminamiento Integrado

Cuando se utilizan una serie de protocolos de encaminamiento en redes multiprotocolo para cada protocolo de red se vuelve ineficiente. Se denomina encaminamiento integrado a la utilización de un solo protocolo de encaminamiento.

Para realizar esto se necesitan realizar algunas modificaciones:

Con formato: Numeración y viñetas

- Codificación de distintos formatos de direcciones
- Información de qué protocolos de red soporta cada Sistema Intermedio
- Información particular de cada protocolo de red

Un ejemplo de encaminamiento es el protocolo IS-IS integrado descrito anteriormente.

Eliminado: 2

3.3. Protocolos de Encaminamiento Interdominio (EGP's)

Los ERP o EGP's (Exterior Routing Protocol o Exterior Gateway Protocol) se usan para intercambiar información de encaminamiento entre diferentes Sistemas Autónomos.

Entre los EGP's más utilizados tenemos: EGP (Exterior Gateway Protocol) y BGP (Border Gateway Protocol), el cual está sustituyendo progresivamente a EGP.

3.3.1. Protocolo EGP

EGP es el protocolo utilizado para el intercambio de información de encaminamiento entre enrutadores exteriores (que no pertenecen al mismo SA). Los enrutadores exteriores sólo pueden retransmitir información de accesibilidad para las redes de su SA. El

Eliminado: 89¶

enrutador debe recoger esta información, habitualmente por medio de un IGP, usado para intercambio entre enrutadores de un mismo SA.

EGP se basa en el sondeo periódico empleando intercambios de mensajes *Hello/I Hera you*, para monitorizar la accesibilidad de los vecinos y para sondear si hay solicitudes de actualización. EGP restringe las pasarelas exteriores al permitirles anunciar sólo las redes de destino accesibles en el SA del enrutador. De esta forma, un enrutador exterior que usa EGP pasa información a sus vecinos EGP pero no anuncia la información de accesibilidad de estos (los enrutadores son vecinos si intercambian información de encaminamiento) fuera del SA. Tiene tres características principales:

- Soporta un protocolo NAP (Neighbor Acquisition Protocol). Dos enrutadores se pueden considerar vecinos si están conectados por una red que es tranparente para ambos. EGP no especifica la forma en que un enrutador decide inicialmente que quiere ser vecino de otro. Para convertirse en vecino, debe enviar un mensaje "*Acquisition confirm*" como respuesta a un "*Acquisition request*". Este paso es necesario para obtener información de encaminamiento de otra pasarela.

Con formato: Numeración y viñetas

- Soporta un protocolo *NR* (*Neighbor Reachability*). El enrutador lo usa para mantener información en tiempo real sobre la accesibilidad de sus vecinos. El protocolo EGP proporciona dos tipos de mensajes para es fin: un mensaje *Hello* y un mensaje *I Hear you (respuesta a Hello).*

Con formato: Numeración y viñetas

- Soporta mensajes de actualización (o mensajes NR) que llevan información de encaminamiento. No se requiere ningún enrutador para enviar mensajes NR u otro enrutador, excepto como respuesta a una petición de sondeo (*poll request).*

Con formato: Numeración y viñetas

Para realizar estas tres funciones básicas, EGP define 10 tipos de mensajes:

Acquisition request: solicita que un enrutador se convierta en vecino.
Acquisition confirm: respuesta afirmativa a un "acquisition request".
Acquisition refuse: respuesta negativa a un "acquisition request".

Eliminado: 89¶

Cease request: solicitud de terminación de la relación de vecindad.

Cease confirm: confirmación para que cesen las peticiones.

Hello: solicitud de respuesta a un vecino, si esta vivo.

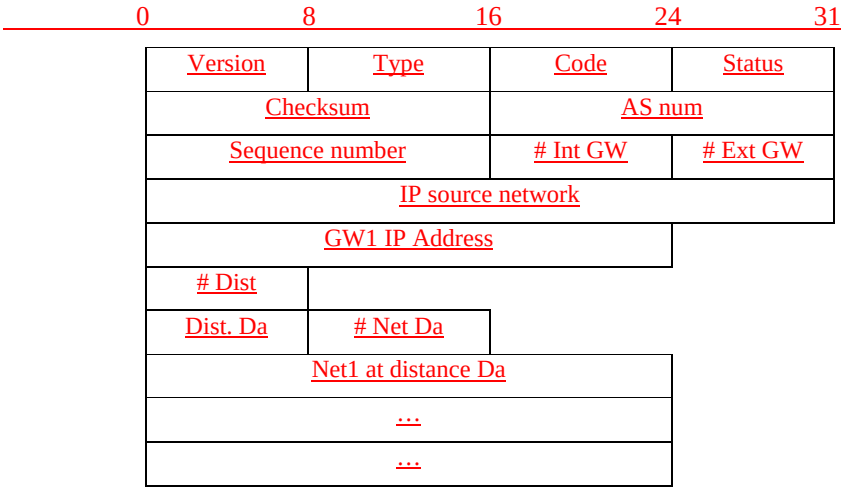
I Hear you: respuesta al mensaje Hello.

Poll request: solicitud de la tabla de encaminamiento de la red.

Routing update: información de accesibilidad de la red.

Error: respuesta a un mensaje incorrecto.

Consideremos el mensaje de actualización, mostrado en la siguiente figura:



En donde los campos son los siguientes (no se consideran los campos de cabecera EGP):

 # Int GW: número de enrutadores interiores que aparecen en el mensaje.

 # Exte GW: número de enrutadores exteriores que aparecen en el mensaje.

IP source network: La dirección IP de red para la que se mide la accesibilidad

GW1 IP address: dirección IP sin el número de red del enrutador para el que se miden las distancias.

 # Dist: número de distancias en el bloque de la pasarela.

Dist. Da: valor de la distancia.

 # Net Da: número de redes a una distancia dada (Da)

Net1 at distance Da: número IP de la red accesible por GW1 a una distancia Da de GW1.

Eliminado: 89¶

Eliminado: ¶

Eliminado: 2

Eliminado: 1

3.3.2. Protocolo BGP

No es posible utilizar un protocolo de encaminamiento intradominio entre Sistemas Autónomos, hace falta utilizar otro tipo de protocolos como los de encaminamiento interdominio (protocolos de pasarela exterior).

Los enrutadores de protocolo de pasarela exterior tienen que preocuparse por los asuntos políticos, y mucho. Por ejemplo, un SA corporativo podría querer ser capaz de enviar paquetes a cualquier instalación Internet y recibir paquetes de cualquier instalación Internet; sin embargo, talvés no quiera transportar paquetes en tránsito que se originan en un SA foráneo y van a otro SA foráneo, aún si su propio SA está en la trayectoria más corta entre los dos SA foráneos (“ese es su problema, no el nuestro”). Por otra parte, el SA podría estar dispuesto a transportar tráfico en tránsito de sus vecinos, o inclusive de otros SA que pagaran por su servicio. Por ejemplo, las compañías telefónicas podrían estar felices de funcionar como portadoras para sus clientes, pero no para otros. Los protocolos de pasarela exterior en general, y en particular del BGP, se han diseñado para permitir muchos tipos de políticas de enrutamiento aplicados al tráfico interSA.

Las políticas típicas comprenden consideraciones políticas, de seguridad o económicas. Algunos ejemplos de restricciones de enrutamiento son:

1. Prohibición del tráfico en tránsito a través de ciertos SA.
2. Nunca poner a Irak en una ruta que comience en el Pentágono.
3. No utilizar a Estados Unidos para llegar de la Columbia Británica a Ontario.
4. Sólo transitar por Albania si no hay una alternativa hacia el destino.
5. El tráfico que comience o termine en IBM no debe transitar por Microsoft.

Las políticas se configuran manualmente en cada enrutador BGP. No son parte del protocolo mismo.

Desde el punto de vista de un enrutador BGP, el mundo consiste en otros enrutadores BGP y en los enlaces que lo conectan. Se consideran conectados dos enrutadores BGP si comparten una red común. Dado el interés especial del BGP en el tráfico de tránsito, las

redes se agrupan en una de tres categorías. La primera categoría es la redes de punta, que sólo tienen una conexión al grafo BGP; no se pueden usar para tráfico en tránsito porque no hay nadie del otro lado. Después vienen las redes multiconectadas. Éstas podrían usarse para el tráfico en tránsito, excepto que se nieguen. Por último, están las redes de tránsito, como los backbones, que están dispuestas a manejar los paquetes de terceros, posiblemente sin algunas restricciones.

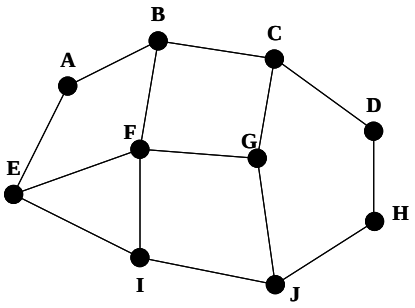
Los pares de enrutadores BGP se comunican entre ellos estableciendo conexiones TCP. Este tipo de operación proporciona comunicación confiable y esconde todos los detalles de la red por la que se pasa.

El BGP fundamentalmente es un protocolo de vector de distancia, pero muy diferente de casi todos los demás, como el RIP. En lugar de mantener sólo el costo a cada destino, cada enrutador BGP lleva el registro de la trayectoria seguida. Del mismo modo, en lugar de dar periódicamente a cada vecino sus costos estimados a todos los destinos posibles, cada enrutador BGP le dice a sus vecinos la trayectoria exacta que está usando.

Como ejemplo, considere los enrutadores BGP mostrados en la figura siguiente. En particular, considere la tabla de enrutamiento de *F*. Suponga que se usa la trayectoria FGCD para llegar a *D*. Cuando los vecinos dan su información de rutas, proporcionan su trayectorias completas.

Información que F recibe
de sus vecinos acerca de D

De B: “Yo uso BCD”
De G: “Yo uso GCD”
De I: “Yo uso IFGCD”
De E: “Yo uso EFGCD”



Una vez que llegan todas las trayectorias de los vecinos, *F* las examina para ver cuál es la mejor. Pronto descarta las trayectorias de *I* y *E*, pues éstas pasan a través de *F* mismo. La decisión entonces está entre usar *B* o *G*. Cada enrutador BGP contiene un módulo que

examina las rutas a un destino dado y las pondera, devolviendo un número para la "distancia" a ese destino por cada ruta. Cualquier ruta que viole una restricción por política automáticamente recibe una ponderación infinita. El enrutador entonces toma la ruta con la distancia más corta. La función de ponderación no es parte del protocolo BGP y puede ser cualquier función que quieran los administradores del sistema.

El BGP soluciona fácilmente el problema de conteo a infinito que es la plaga de otros algoritmos de enrutamiento por vector de distancia. Por ejemplo, supóngase que se cae *G* o se desactiva la línea *FG*. Entonces *F* recibe rutas de sus tres vecinos restantes. Estas rutas son *BCD, IFGCD y EFGCD*. Puede verse de inmediato que las dos últimas rutas no tienen sentido, pues pasan a través del mismo *F*, por lo que se escoge *FBCD* como ruta nueva. Otros algoritmos de vector de distancia con frecuencia toman las decisiones equivocadas porque no pueden saber cuáles de sus vecinos tienen rutas independientes a los destinos, y cuáles no.

Existen cuatro versiones de BGP. Cuando se dice BGP, suele hacerse referencia a la versión 3, a menos que se trate de un documento anterior a esta versión.

BGP-3 es un protocolo de encaminamiento interSA basado en la experiencia obtenida de EGP. A diferencia de otros protocolos de encaminamiento que se comunican mediante paquetes o datos, BGP-3 está orientado a conexión; utiliza TCP como protocolo de transporte. El número de puerto es 179.

Recuérdese que EGP se diseñó como un protocolo para intercambiar información de encaminamiento entre SA's, mas que como un verdadero protocolo de encaminamiento. Debido a que la información de encaminamiento interSA no está disponible, EGP no puede detectar la presencia de un bucle causado por un conjunto de enrutadores que creen que uno de ellos puede alcanzar otro SA al que ninguno de ellos está conectado. Un problema adicional con EGP tiene que ver con la cantidad de información intercambiada; a medida que el número de redes IP que conoce NSFnet aumenta, el tamaño de los mensajes NR aumenta también y la cantidad de tiempo necesario para procesarlos se hace significativa.

Eliminado: 89¶

BGP-3 ha sustituido a EGP en la troncal NSFnet por estas razones. Sin embargo BGP-3, no requiere que NSFnet o cualquier otra troncal juegue un papel central, en comparación con el carácter de núcleo que jugó ARPANET en los primeros tiempos de Internet. En vez de eso, BGP-3 ve Internet como una colección de SA's y no tiene en cuenta la topología interna de un SA ni el IGP que utilice.

Ahora definiremos algunos términos utilizados en BGP-3:

BGP speaker (BGPS): Un sistema que ejecuta BGP.

BGP neighbors: Un par de BGPs intercambiando información de encaminamiento interSA. Los vecinos BGP pueden ser de dos tipos:

Con formato

Internal: Un par de BGP en el mismo SA.

External: Un par de BGP en diferentes SA.

BGP session: Una sesión BGP entre vecinos BGP que se intercambian información de encaminamiento por medio de BGP. Los vecinos monitorizan el estado de la conexión enviando un mensaje "keepalive" regularmente (intervalo recomendado de 30 seg.)

AS Border Router (ASBR): Un router con conexión a múltiples SA. Existen dos tipos de ASBR, dependiendo de su relación topológica con el BGPS al que se refieren.

Internal: Un enrutador a un salto de distancia en el mismo SA que el BGPS.

External: Un enrutador a un salto de distancia en distinto SA que el BGPS.

La dirección IP de un ASBR se especifica como el siguiente salto cuando BGP-3 anuncia una ruta SA a uno de sus vecinos. Dicho ASBR debe compartir una conexión física con los BGPSs emisor y receptor. Si un BGPS detecta un ASBR como siguiente salto, el BGPS debe conocerlo previamente por sus sondeos.

AS connection: BGP-3 define dos tipos de conexión interSA

Physical connection: Un SA comparte una red física con otro SA, y esta red está conectada a al menos un ASBR de cada SA. Como estos dos ASBR comparten una red, se pueden enviar paquetes sin requerir ningún protocolo de encaminamiento interSA o intraSA (es decir, no requieren de IGP ni de EGP para comunicarse).

Con formato

Con formato

BGP connection: Una conexión BGP significa que hay una sesión BGP entre un par de BGPSs, uno en cada SA, y esta sesión se usa para comunicar las rutas a través de los ASBRs que se pueden emplear para redes específicas. El término conexión BGP se puede usar para referirse a una sesión entre dos BGPS en el mismo SA.

Traffic type: BGP-3 categoriza el tráfico en un SA en dos tipos.

Local: El tráfico que se origina o que termina en ese SA. Es decir, o bien la dirección fuente o bien la de destino están en el SA.

Transit: Tráfico no local.

Uno de los objetivos de BGP es minimizar este tipo de tráfico.

AS Type: Un SA se clasifica en uno de tres tipos.

Stub: Un SAS (stub AS) tiene una sola conexión interSA con otro SA, y solo lleva tráfico de tipo *local*.

Multihomed: Un SA multihomed (multipuerto) tiene conexiones a uno o más SA pero rechaza llevar tráfico de tipo *transit.*

Transit: Un SA de tipo transit tiene conexiones a uno o más SA y lleva tráfico de tipo transit. El SA puede imponer restricciones en el tráfico que llevará.

AS number: Un número de 16 bits que identifica unívocamente el SA. Es el mismo número que usan GGP y EGP.

AS path: Una lista de todos los números SA que atraviesa una ruta al intercambiar información de encaminamiento. Más que intercambiar simples valores de métrica, BGP-3 comunica rutas enteras a sus vecinos.

Routing Policy: Un conjunto de reglas que constriñen el encaminamiento para adecuarse a los deseos de la autoridad que administra el AS. Las políticas de encaminamiento no están definidas en el protocolo BGP-3, pero están seleccionadas por la autoridad AS y se presentan a BGP-3 en forma de datos de configuración específicos de la implementación. Las políticas de encaminamiento las puede seleccionar la autoridad del AS del modo que considere oportuno. Por ejemplo:

- Un SA "multihomed" puede rechazar actuar como SA "transit". Lo consigue no anunciándose a otras redes más que a las conectadas directamente con él.
- Un SA "multihomed" puede limitarse a ser de tipo "transit" para un número restringido de SAs adyacentes. Lo consigue anunciando su información de encaminamiento sólo a este conjunto.

Eliminado: 89¶

Con formato

Con formato

Eliminado: ¶

Con formato

Con formato: Numeración y viñetas

- Un SA puede seleccionar que SA externo se debería usar para llevar tráfico de tipo "transit". También puede aplicar criterios de rendimiento al seleccionar las rutas al exterior:
  - Un AS puede optimizar el tráfico para usar rutas cortas.
  - Un AS puede seleccionar rutas de tránsito según la calidad el servicio en los saltos intermedios. Esta calidad se podría determinar con mecanismos ajenos a BGP-3.

De la definición anterior se puede ver que un SAS o un SA "multihomed" tienen las mismas propiedades topológicas que en la arquitectura ARPANET: nunca actúan como SA intermedios (intermediate AS) en una ruta interSA. En la arquitectura ARPANET, EGP bastaba para que esta clase de SAs intercambiase información de accesibilidad con sus vecinos, y esto sigue siendo cierto con BGP-3. Por tanto, un SAS o un SA "multihomed" pueden seguir usando EGP (o cualquier otro protocolo adecuado) para operar con un SA de tipo "transit". Sin embargo, es recomendable usar BGP. Adicionalmente, en un SA "multihomed", es probable que BGP proporcione un encaminamiento interAS más óptimo que EGP, ya que EGP no considera la distancia.

3.3.2.1. Selección de la ruta

Cada BGPS debe evaluar distintas rutas a un destino desde los ASBRs de la conexión con el SA, seleccionar la que mejor cumpla la política de encaminamiento y luego anunciar esa ruta a todos sus vecinos en la conexión con el SA.

BGP-3 es un protocolo vector-distancia pero, a diferencia de los protocolos vector-distancia tradicionales tales como RIP, en los que existe sólo una métrica, BGP determina un orden de preferencia al aplicar una función que mapea cada ruta a un valor de prioridad y selecciona la ruta que tanga el mayor valor. Esta función la genera la implementación de BGP-3 según la información de configuración.

Eliminado: 89¶

Cuando hay múltiples rutas hasta un destino, BGP-3 las mantiene todas pero sólo anuncia la de mayor preferencia. Esta estrategia permite cambiar rápidamente a una ruta alternativa cuando falla la principal.

Con formato: Numeración y viñetas

3.3.2.2. Políticas de encaminamiento

- Una implementación de BGP-3 debería ser capaz de controlar las rutas que anuncia. La granularidad de este control debería estar al menos al nivel de red para las rutas anunciadas y al del SA para los receptores.
- BGP-3 debería permitir una política de ponderación para las rutas. A cada SA se le puede asignar un peso específico de modo que la ruta preferida a un destino es la de menor peso resultante de la agregación de los pesos de los Sas.
- BGP-3 debería permitir una política de exclusión de un SA de todas las posibles rutas . Esto se puede hacer con una variante de la política anterior; a cada SA a excluir se le da un peso “infinito” y el proceso de selección de rutas se encargará de rechazar las rutas de peso infinito.

3.3.2.3. Consistencia de un AS

BGP-3 requiere que un SA tipo "transit" presente el mismo aspecto a todo SA que emplee sus servicios. Si el SA tiene múltiples BGPSs, deben estar de acuerdo sobre dos aspectos de la topología: intraSA e interSA. Como BGP no maneja el encaminamiento intraSA en absoluto, el protocolo de encaminamiento interior debe dar una visión consistente de la topología intraSA. Naturalmente, un protocolo tal como OSPF o IS-IS Integrado que implementa la sincronización de bases de datos de enrutadores se presta por sí misma a este papel. La consistencia de la topología eterna la *pueden* proporcionar todos los BGPSs del SA que tengan sesiones entre sí, pero BGP-3 no pide que se utilice este método, sólo que se mantenga la consistencia.

3.3.2.4. Intercambio de información de encaminamiento

BGP-3 sólo anuncia las rutas usadas con sus vecinos. Es decir, se adapta al paradigma habitual salto-a-salto de Internet, incluso si tiene información adicional en la forma de rutas SA y aunque fuese capaz de informar a un vecino de una ruta que él mismo no usa.

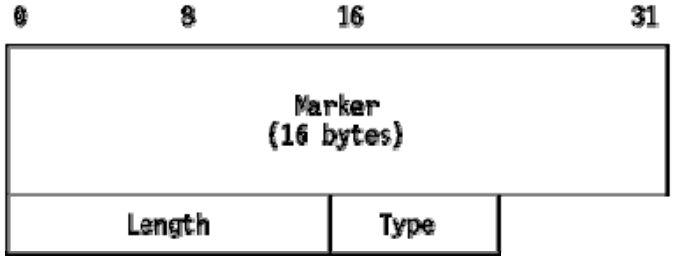
Eliminado: 89¶

Cuando dos BGPSs forma una sesión BGP, comienzan a intercambiar todas sus tablas de encaminamiento. La información de encaminamiento se intercambia por medio de mensajes UPDATE. Como la información de encaminamiento contiene la ruta completa para cada destino en forma de una lista de números de SA además de la información normal de accesibilidad y del siguiente salto empleadas en protocolos vector-distancia, se puede usar para eliminar los bucles y para eliminar el problema de la cuenta hasta infinito de RIP. Después de que los vecinos han efectuado su intercambio inicial de sus bases de datos, sólo se envían actualizaciones de esa información.

Con formato

3.3.2.5. Formato de mensaje de lBGP-3

Todos los mensajes BGP-3 tienen en común un formato básico. Varían en longitud de 19 a 4096 bytes, se transmiten sobre TCP y se procesan en su totalidad(no se procesan hasta que se han recibido por completo). Cada mensaje tiene una cabecera mostrada en la siguiente figura (cabecera BGP-3).



Marker: Un valor que el receptor puede predecir, usado para la autentificación y para identificar pérdidas de sincronización. Se rellena con unos cuando " Authentication Code" es 0.

Con formato

Length: Longitud total del mensaje, incluyendo la cabecera, en bytes. El mensaje no se puede rellenar o engordar ya que en muchos casos la longitud se utiliza para calcular la longitud del último campo del mensaje.

Con formato

Type: Un valor sin signo de 8 bits.

1 OPEN (10)

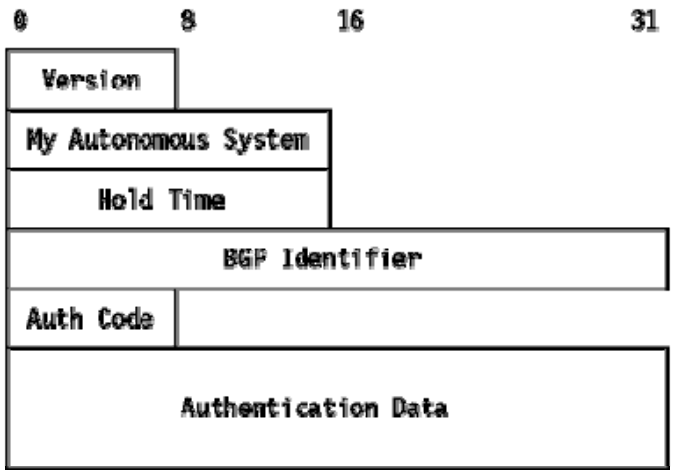
2 UPDATE

3 NOTIFICATION

Eliminado: 89¶

4 KEEPALIVE

Los mensajes OPEN se usan para iniciar la sesión BGP-3. El formato se muestra en la siguiente figura (Mensaje OPEN de BGP-3).



Version: 3 para BGP-3(1 byte)

Con formato

My Autonomous System: El número de SA del emisor (2 bytes) .

Con formato

Hold time: El tiempo máximo en segundos que puede transcurrir entre la recepción de sucesivos mensajes KEEPALIVE y/o UPDATE y/o NOTIFICATION (2 bytes).

Con formato

BGP Identifier: Un número de 32 bits único que identifica al BGPS. Es la dirección IP de cualquiera de sus interfaces. Se usa el mismo número para todas las interfaces y vecinos BGP.

Con formato

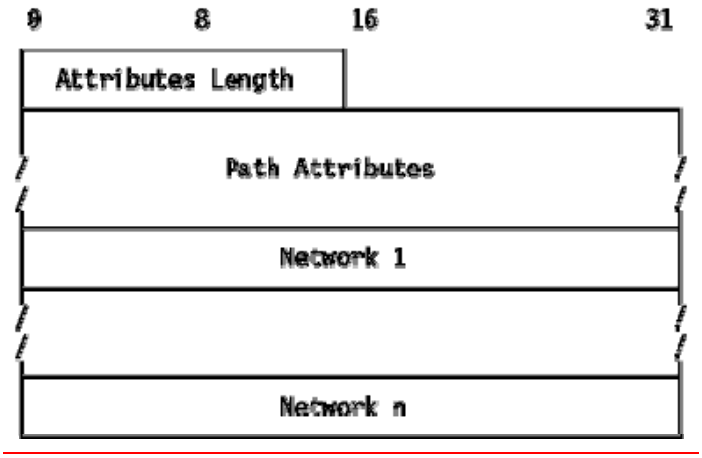
Authentication Code: Define la interpretación de "Authentication Data" (1 byte). BGP-3 sólo define el código de autentificación 0 (no hay autentificación).

Con formato

Authentication Data: Dependiente de "Authentication Code". La longitud es variable y se deduce a partir de la longitud del mensaje. Para el código 0, el dato se omite.

Los mensajes UPDATE se emplean para transmitir información de encaminamiento. El formato de un mensaje UPDATE se muestra en la figura a continuación (mensaje update de BGP-3).

Eliminado: 89¶



Attributes Length: Longitud del campo "path attributes" bytes (2 bytes).

Path Attributes: Cada path attribute (atributo de la ruta) es una tripleta: < attribute type, attribute length, attribute value> donde:

attribute type: es una campo de 2 bytes, consistente en una byte de flag y un byte de código del tipo de atributo. Los bits del byte de flag son:

X'80': Atributo opcional. Si está a uno, el atributo es "optional", si no es bien conocido ("well-known"). Los atributos bien conocidos son los que todas las implementaciones de BGP-3 deben manejar. Los hay de dos tipos: "mandatory", que se deben incluir en cada mensaje UPDATE, y "*discretionary*" que se pueden omitir de los mensajes UPDATE. Si un BGPs no reconoce un atributo opcional, debería manejarlo según el bit "transitive". Los BGPs pueden actualizar los atributos de los mensajes que retransmiten.

X'40': Atributo de tipo "transitive"(transitivo). Debe estar a uno si el atributo es de tipo "mandatory". Para atributos opcionales, si este bit está a uno, el atributo es de tipo "*transitive*", si no es de tipo "*non-transitive*". Un atributo "transitive" no reconocido se debe pasar a las consultas de otro BGP después de poner el bit "partial" a uno, y puede ser desechado. Los BGPSs pueden añadir atributos "transitive" de tipo "optional" en un mensaje UPDATE antes de retransmitirlo.

Con formato
Con formato
Con formato
Con formato
Con formato
Con formato
Con formato
Con formato
Con formato

Eliminado: 89¶

X'20': Atributo de tipo "partial". Este bit indica que se pasó un atributo "optional" y "transitive" a un BGPS que no lo reconoció o que fue añadido por un BGPS distinto del emisor. En todos los demás casos debe ser cero.

Con formato

X'10': "Extended length". El campo "attribute length" consta de dos bytes si este bit vale 1, y de uno si es 0. Los cuatro bits de orden inferior son cero y el receptor debe ignorarlos.

Con formato

Los valores "attribute type code" (códigos del tipo de atributo) se muestran en la siguiente tabla:

Con formato

ORIGIN: El método por el que el AS emisor conoció esta ruta.

0 IGP -- las redes listadas están dentro del AS emisor.

1 EGP -- las redes listadas están fuera del AS emisor y la información de accesibilidad se consiguió por EGP.

2 INCOMPLETE -- las redes listadas se conocieron por otros medios.

AS_PATH: Los números AS de 2 bytes de cada AS en la ruta a la red/es de destino. El número de saltos en la ruta se puede calcular dividiendo al campo "attribute length" por 2.

NEXT_HOP: La dirección IP del ABR que es el siguiente salto en la ruta a la red/es listada/s. Este campo se ignora para las conexiones BGP internas.

UNREACHABLE: Las rutas anunciadas previamente se han convertido en inalcanzables.

INTER-AS METRIC: Este valor se puede usar para elegir entre múltiples rutas a un AS. Si los demás factores son iguales, se elige la ruta con métrica más baja. Este valor se le puede enviar a un BGPS en un AS vecino, y si se recibe sobre una conexión BGP, se puede propagar a través de conexiones BGP internas. Un BGPS no puede retransmitir un INTER-AS METRIC en un mensaje UPDATE al exterior.

attribute length: Longitud(uno o dos bytes).

Con formato

attribute value: Dependiente del código del valor "attribute type code".

Con formato

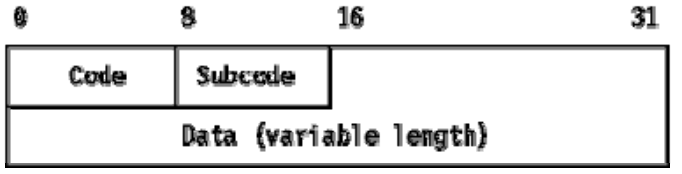
Cada "attribute"(atributo) sólo se puede especificar una vez. El campo "attribute length" determina dónde acaban.

Eliminado: 89¶

Network 1: El número de red de 32 bits de la primera red descrita en los "path attributes" anteriores. Las subredes y los hosts están inhabilitados explícitamente.

Network n: El número de red de 32 bits de la última red descrita en los "path attributes" anteriores. Las subredes y los hosts están inhabilitados explícitamente. El número de red se puede calcular restando las longitudes de la cabecera BGP-3 y del campo "path attributes" a la longitud del mensaje y dividiendo por 4.

Los mensajes NOTIFICATION se usan para informar al vecino de un error. La conexión BGP se termina tras enviar el mensaje. El formato de este mensaje se muestra a continuación.



Code: Un byte indicando el tipo de error. Están definidos los siguientes códigos:

1 Message Header Error
2 OPEN Message Error
3 UPDATE Message Error
4 Hold Timer Expired
5 Finite State Machine Error
6 Cease

Subcode: Un byte de subcódigo que proporciona más información acerca del error. El valor 0 indica que no existen un valor adecuado para este campo. Están definidos los siguientes subcódigos:

Message Header Error Subcodes

1 Connection Not Synchronized
2 Bad Message Length
3 Bad Message Type

OPEN Message Error Subcodes

1 Unsupported Version Number
2 Bad Peer AS

Eliminado: 89¶

3 Bad BGP Identifier

4 Unsupported Authentication Code

5 Authentication Failure

UPDATE Message Error Subcodes

1 Malformed Attribute List

2 Unrecognized Well-known Attribute

3 Missing Well-known Attribute

4 Attribute Flags Error

5 Attribute Length Error

6 Invalid ORIGIN Attribute

7 AS Routing Loop

8 Invalid NEXT_HOP Attribute

9 Optional Attribute Error

10 Invalid Network Field

Data: Información de longitud variable dependiente del código y del subcódigo que se pueden emplear para diagnosticar la causa del error. La longitud se puede calcular sustrayendo 21 a la longitud total del mensaje.

Con formato

Los mensajes KEEPALIVE se emplean para asegurarse de que la conexión sigue activa. Consisten sólo en la cabecera.

Eliminado: 2.3.1.1. Protocolo IBGP¶
¶
2.3.1.2. Protocolo EBGP¶
¶
¶
¶
¶
¶

Con formato

Eliminado: 89¶

Conclusiones

- El estudio de encaminamiento en redes IP, es una parte muy importante en el diseño de redes, ya que es en el nivel de red en donde se abordan los problemas y protocolos necesarios para llevar a cabo la conexión de dos redes distintas.

- Los Algoritmos de enrutamiento deben ser bien estudiados para su implantación en los enrutadores. El encaminamiento en un enrutador con algoritmos equivocados puede provocar retardos y pérdidas de datos antes que soluciones.

- Faltan por revisar los temas del control de congestión y control de flujo (no fueron parte del curso), que provoca degradación en el desempeño de la subred. Hay que analizar cuáles son aquellos algoritmos y bajo que circunstancias provocan congestionamiento y un tráfico muy elevado.

Eliminado: ¶
¶
¶
Salto de página
Indice¶
¶

... [3]

Eliminado: 89¶

Recomendaciones

- En lo posible, a todos los enrutadores utilizados para encaminamiento interdominio se los debe configurar con el protocolo BGP y no con EGP, que aún presenta inconvenientes, además de ser antiguo .

Con formato: Numeración y viñetas

- Implementar el protocolo OSPF (aparte de los otros protocolos necesarios) en los enrutadores para el encaminamiento intradominio, que presenta muchas ventajas con respecto a los otros tipos de protocolos de encaminamiento interior.

Con formato: Numeración y viñetas

- Con la llegada de IPV6, las direcciones de Internet deben ser organizadas de una manera jerárquica de tal manera las tablas de encaminamiento en los enrutadores sean lo más pequeñas posible y no causen retardos en la transmisión de paquetes.

Con formato: Numeración y viñetas

- Una recomendación muy especial a la Universidad del Azuay: realizar prácticas sobre configuración de enrutadores, ya que sin la aplicación real, todos los conocimientos teóricos adquiridos no pueden ser llevados a la práctica y pierden su verdadero valor.

Con formato

Con formato: Numeración y viñetas

Con formato

Eliminado: 89¶

Bibliografía

- TANENBAUM, Andrew S., "Redes de Computadoras", Tercera Edición, Editorial Prentice Hall Hispanoamericana, S.A. México, 1997
  ISBN 968-880-958-6

- CETTICO (Centro de Transferencia Tecnológica en Informática y Comunicaciones), "Curso de Informática Personal, Teleinformática", Edición 1999, Editorial Cultural, Madrid-España, 1999
  ISBN 84-8055-275-1 obra completa
  ISBN 84-8055-277-8

- BASSAM, Halabi, "Internet Routing Architectures", Cisco Press, 2000
  ISBN 157870233X

- MOY, John, "OSPF: Anatomy of an Internet Ruting Protocol", Tercera Edición, Addison Wesley, 1998

- Material entregado en el Postgrado "Ingeniaría de Sistemas Internet y Móviles", Universidad Politécnica de Madrid, Octubre – Diciembre 2002. Madrid

Con formato: Numeración y viñetas

Con formato: Numeración y viñetas

Con formato

Con formato: Numeración y viñetas

Con formato: Numeración y viñetas

Con formato: Numeración y viñetas

Consultas en Internet

http://www.cisco.com/warp/public/104/1.html

http://www.ittc.ku.edu/EECS/EECS_800.ira/bgp_tutorial/

http://www.research.att.com/~griffin/interdomain.html

http://ingenet.ulpgc.es/~ablesa/telecom/optimizaredes/routing4.htm

http://ditec.um.es/laso/docs/tut-tcpip/3376ch3.html

http://gsyc.escet.urjc.es/docencia/asignaturas/redes-I/transparencias/routing/routing.html

Código de campo

Código de campo

Código de campo

Código de campo

Código de campo

Código de campo

Con formato

GLOSARIO

AAL1	ATM Adaptation Layer 1.- Nivel de adaptación 1 en ATM. Protocolo utilizado para transmitir tráfico orientado a conexiones de tiempo real y con tasa de bits constante.
Algoritmo	Secuencia ordenada de instrucciones que permite resolver un problema planteado.
Ancho de banda	Es un rango de frecuencias dentro de las cuales se realiza la transmisión de datos.
Arquitectura de red	Se refiere a la especificación funcional del sistema y sus componentes. Esta especificación no describe cómo hay que implementar la arquitectura, sino describe los elementos de la misma y su disposición.
ATM	Asyncronous Transference Mode.- Modo de transferencia asíncrona. Tecnología de transmisión de datos que utiliza celdas para el envío de información.
Buffers	Memoria de acceso rápido
Calidad de servicio	QoS.- se refiere a los parámetros que se definen para establecer la calidad en la transferencia y entrega de los datos.
Capa de enlace	Este nivel detecta, y posiblemente corrige, los errores que ocurren en el nivel físico, y así proporciona una línea libre de errores de transmisión al nivel de red.
Capa de transporte	Este nivel asegura que las unidades de datos son entregadas sin errores, en orden y sin duplicación ni pérdidas. También está relacionado con la optimización del uso de los servicios de red y el ofrecimiento de servicios de control de parámetros de comunicación.
Conmutación de circuitos	En una transmisión utilizando la conmutación de circuito, se establece un circuito virtual entre los hosts que se comunican, y permanece abierto el circuito mientras dura la transmisión luego de la cual se cierra dicho circuito virtual.
Conmutación de paquetes	En una transmisión utilizando la conmutación de paquetes, cada uno de los paquetes de datos se dirige desde el origen hasta el destino por el “mejor camino” a través de cualquier nodo intermedio dentro la red.
Control de flujo	Control del trafico de transmisión de datos, para que no exista congestionamiento y se pierdan los paquetes.
Datagrama	Paquetes independientes de tipo sin conexiones
Detección de errores	Detectar errores en la transmisión de datos, para poder retransmitirlos o descartarlos.
Enrutador	Es un equipo de comunicación que encamina los paquetes de datos desde el origen hasta un destino preestablecido.
Enrutar	Encaminar paquetes desde el origen hasta el destino, directo o a través de nodos intermedios.
Entidad	Es cada uno de los elementos dentro de una red que emiten o reciben datos.
Ethernet	Es un modo de conexión (ver topología) de una LAN, en la cual

	existe un medio común al cual se conectan cada una de los equipos. Es una topología en bus
Grafo	Gráfico utilizado para representar una red o una subred. En un grafo se utilizan nodos y enlaces para representar enrutadores y líneas de datos respectivamente.
Host	Es una máquina dedicada a ejecutar programas de usuario (programas de aplicación).
Internet	Contracción de INTERnational NETwork. Es la gran red de redes, a la cual se encuentran conectados millones de usuarios en todo el mundo.
IP	Internet Protocol.- Protocolo de Internet. También se define paquetes Internet, en los cuales se inserta los datos para ser transmitidos.
LAN	Local Area Network.- Red de área local, tiene una cobertura de hasta unos 10 km.
Niveles de red	Cada una de las capas en las que se ha dividido un arquitectura de redes. En el modelo OSI existen definidos 7 niveles o capas de red.
Nodo	Utilizado para representar un enrutador en los grafos.
OSI	Open System Interconnection.- Interconexión de Sistemas Abiertos. Es un modelo de referencia estándar para representar la arquitectura de redes.
Paquetes	Es un grupo de datos, generalmente del mismo tamaño, que se transmite desde una máquina origen hacia una o varias máquinas destino.
Path	Camino.- es la trayectoria que se sigue para llegar de un lugar a otro. Utilizando en una PC, path significa la ubicación de un archivo o carpeta en un dispositivo de almacenamiento.
Servicio de petición y respuesta	En este servicio el remitente transmite un datagrama sencillo que contiene una petición; la respuesta contiene la contestación del receptor.
Servicios no orientados a conexión	Cada mensaje lleva la dirección completa del destino, y cada uno se encamina a través del sistema de forma independiente de todos los demás.
Servicios orientados a conexión	El origen establece primero una conexión con el destino, luego la usa y después la libera.
Subred	Una subred es una porción de la red, que conduce mensajes de una host a otra.
TCP/IP	Transmission Control Protocol/Internet Protocolo.- Protocolo de control de transmisión / Protocolo de Internet. Conjunto de protocolos utilizados en Internet.
Topología	Es la manera cómo están conectados los cables de una red.
UDP	User Datagram Protocol.- Protocolo de datagrama de usuario, es un protocolo sin conexión no confiable, para aplicaciones que no necesitan la asignación de secuencia ni el control de flujo del TCP y que desean utilizar las suyos propios.
WAN	Wide Area Network.- Red de Área Extensa.

Eliminado: 89¶

Anexo 1

Algoritmo de Dijkstra

Eliminado: Paquete Hello¶ Paquete ECHO¶

Con formato

El Algoritmo de Dijkstra es utilizado para calcular la trayectoria más corta entre dos nodos de un grafo. En el programa (escrito en lenguaje C) a continuación, el cálculo se hace comenzando por el nodo terminal, ***t***, en lugar del nodo de origen, ***s***. Dado que la trayectoria más corta posible de ***t*** a ***s*** en un grafo no dirigido es igual a la trayectoria más corta de ***s*** a ***t***, no importa el extremo por el que comencemos (a menos que haya varias trayectorias más cortas posibles, en cuyo caso la inversión de la búsqueda podría descubrir una distancia). La razón de una búsqueda en reversa es que cada nodo está etiquetado con su antecesor, en lugar de su sucesor. Al copiar la trayectoria final en la variable de salida, ***path***, la trayectoria de salida se invierte. Al invertir la búsqueda, ambos efectos se cancelan, y la respuesta se produce en el orden correcto.

Con formato

Con formato

Con formato

Con formato

Con formato

Con formato

Con formato

```
#define MAX_NODES 1024                  /*número máximo de nodos*/
#define INFINITY 1000000000             /*un número mayor que cualquier trayectoria máxima*/
int n, dist[MAX_NODES][MAX_NODES];  /*dist[i][j] es la distancia entre i y j*/

void shortest_path(int s, int t, int path[])
{struct state{                          /*la trayectoria con la que se está trabajando*/
    int predecesor;                     /*nodo previo*/
    int lenght;                         /*longitud del origen a este nodo*/
    enum {permanent, tentative} label;  /*estado de la etiqueta*/
} state[MAX_NODES];

int i, k, min;
struct state *
          p;
for (p = &state[0]; p < &state[n]; p++) {     /*estado de inicialización*/
    p->predecesor = -1;
    p->lenght = INFINITY;
    p->label = tentative;
}
state[t].length = 0; state[t].label = permanent;
```

Eliminado: 89¶

```
k = t;                                          /*k es el nodo de trabajo inicial*/
do {                                            /*¿hay una trayectoria mejor desde k?*/
    for (i = 0; i < n; i++)                     /*este grafo tiene n nodos*/
        if (dist[k][i] !=0 && state[i].label == tentative) {
            state[i].predecesor = k;
            state[i].length = state[k].length + dist[k][i];
        }
    }

    /*Encuentra el nodo etiquetado tentativamente con la etiqueta menor*/
    k = 0; min = INFINITY;
    for (I = 0 < n; i++)
        if (state[i].label == tentative && state[i].lenght < min) {
            min = state[i].lenght;
            k = i;
        }
    state[k].label = permanent;
} while (k != s);

/*copia la trayectoria en el arreglo de la salida*/
i = 0 ; k = s
do {path[i++] = k; k = state[k].predecesor; } while (k>=0);
}
```

Con formato

Eliminado: 89¶

Anexo 2

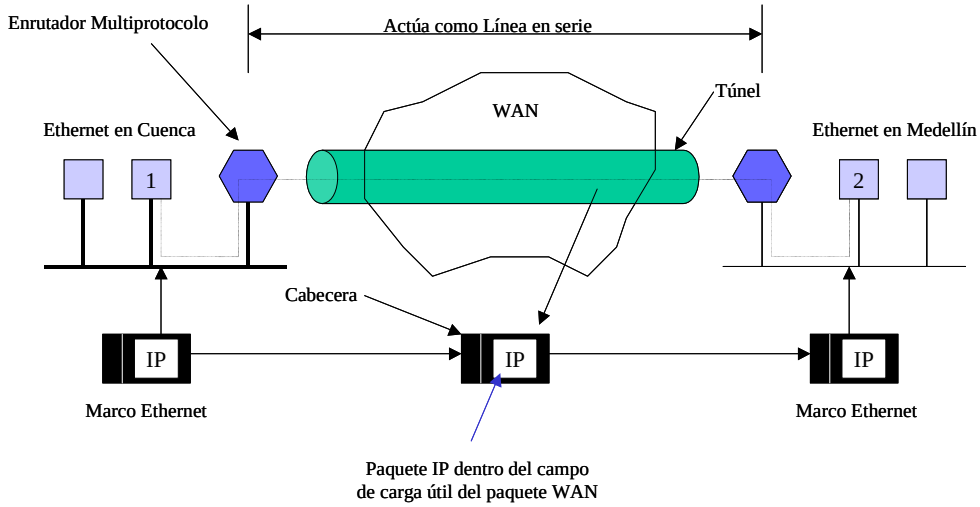
Tuneling

Con formato

Eliminado: ú

Con formato

El manejo del caso general de lograr la interacción de dos redes diferentes es extremadamente difícil. Sin embargo, hay un caso especial común que puede manejarse. Este caso es cuando el host de origen y el de destino están en la misma clase de red, pero hay una red diferente en medio. Como ejemplo, piense en una empresa multinacional con una Ethernet basada en TCP/IP en Cuenca. Una Ethernet basada en TCP/IP en Medellín y una WAN PTT en medio, como se puede apreciar en la siguiente figura:



La solución a este problema es una técnica llamada **proceso de túnel (tuneling)**. Para enviar un paquete IP al *host 2*, el *host 1* construye el paquete que contiene la dirección IP del host 2, lo inserta en un marco Ethernet dirigido al enrutador multiprotocolo de Cuenca, y lo pone en el Ethernet. Cuando el enrutador multiprotocolo recibe el marco, retira el paquete IP, lo inserta en el campo de carga útil del paquete de capa de red de la WAN, y dirige este último a la dirección de la WAN del enrutador multiprotocolo de Medellín. Al llegar ahí, el enrutador de Medellín retira el paquete IP y lo envía al host 2 en un marco Ethernet.

La WAN puede visualizarse como un gran túnel que se extiende de un enrutador multiprotocolo al otro. El paquete IP simplemente viaja de un extremo del túnel al otro, bien

Eliminado: 89¶

acomodado en una “caja bonita”. No tiene que preocuparse por lidiar con la WAN. Tampoco tiene que hacerlo los hosts de cualquiera de los Ethernet. Solo el enrutador multiprotocolo tiene que entender los paquetes IP y WAN. De hecho, la distancia completa entre la mitad de un enrutador multiprotocolo y la mitad del otro actúa como una línea en serie.

Eliminado: 89¶

INDICE
ENCAMINAMIENTO EN REDES IP

Eliminado: 89¶

Eliminado: ... [4]

Con formato

INDICE

ENCAMINAMIENTO EN REDES IP

Salto de página

Página 52: [2] Eliminado **Pablo Durazno** **08/02/2003 23:14:00**

The maximum cost allowed in RIP is 16 which means that the network is unreachable. Thus RIP is inadequate for large networks (that is, those in which legitimate hop counts approach 16).

RIP does not support variable length subnet masks (variable subnetting). There is no facility in a RIP message to specify a subnet mask associated with the IP address.

RIP has no facilities to ensure that routing table updates come from authorized routers. It is an unsecure protocol.

RIP only uses fixed metrics to compare alternative routes. It is not appropriate for situations where routes need to be chosen based on real-time parameters such as measured delay, reliability, or load.

The protocol depends upon counting to infinity to resolve certain unusual situations. As described earlier (Vector-Distance), the resolution of a loop would require either much time (if the frequency of updates was limited) or much bandwidth (if updates

were sent whenever changes were detected). As the size of the routing domain grows, the instability of the vector-distance algorithm in the face of changing topology becomes apparent. RIP specifies mechanisms to minimize the problems with counting to infinity (these are described below) which allows RIP to be used for larger routing domains, but eventually RIP will be unable to cope. There is no fixed upper limit, but the practical maximum depends upon the frequency of changes to the topology, the details of the network topology itself, and what is deemed as an acceptable maximum time for the routing topology to stabilize.

Solving the counting to infinity problem is done by using the split horizon, poisoned reverse and triggered updates techniques.

Split horizon with poisoned reverse

Consideremos nuestra red de ejemplo mostrada en la figura:

As described in Vector-Distancia the problem was caused by the fact that A and C are engaged in a pattern of mutual deception. Each claims to be able to reach D via the other. This can be prevented by being more careful about where information is sent. In particular, it is never useful to claim reachability for a destination network to the neighbor from which the route was learned (reverse routes). The split horizon with poisoned reverse scheme includes routes in updates sent to the router from which they were learned, but sets their metrics to infinity. If two routers have routes pointing at each other, advertising reverse routes with a metric of 16 will break the loop immediately. If the reverse routes are simply not advertised (this scheme is called simple split horizon), the erroneous routes will have to be eliminated by waiting for a timeout. Poisoned reverse does have a disadvantage: it increases the size of the routing messages.

Triggered updates

Split horizon with poisoned reverse will prevent any routing loop that involves only two gateways. However, it is still possible to end up with patterns in which three routers are engaged in mutual deception. For example, A may believe it has a route through B, B through C, and C through A. This cannot be solved using split horizon. This loop will only be resolved when the metric reaches infinity and the network or host involved is then declared unreachable. Triggered updates are an attempt to speed up this convergence. Whenever a router changes the metric for a route, it is

required to send update messages almost immediately, even if it is not yet time for one of the regular update messages (RIP specifies a small time delay, between 1 and 5 seconds, in order to avoid having triggered updates generate excessive network traffic).

Protocolo de información de enrutamiento versión 2 (RIP-2)

RIP-2 es un protocolo estándar borrador. Su estado es electivo y se describe en el RFC 1723.

RIP-2 extende RIP-1. Es menos potente que otros IGPs recientes tales como OSPF (ver Open Shortest Path First Protocol (OSPF) Version 2) and IS-IS (see OSI Intermediate System to Intermediate System (IS-IS)), but it has the advantages of easy implementation and lower overheads. The intention of RIP-2 is to provide a straightforward replacement for RIP which can be used on small to medium-sized networks, can be employed in the presence of variable subnetting (see Subnets) or supernetting (see Classless Inter-Domain Routing (CIDR)) and importantly, can interoperate with RIP-1.

RIP-2 takes advantage of the fact that half of the bytes in a RIP-1 message are reserved (must be zero) and that the original RIP-1 specification was well designed with enhancements in mind, particularly in the use of the version field. One notable area where this is not the case is in the interpretation of the metric field. RIP-1 specifies it as being a value between 0 and 16 stored in a four-byte field. For compatibility, RIP-2 preserves this definition, meaning that it agrees with RIP-1 that 16 is to be interpreted as infinity, and wastes most of this field.

Nota: Neither RIP-1 nor RIP-2 are properly suited for use as an IGP in an AS where a value of 16 is too low to be regarded as infinity, because high values of infinity exacerbate the counting to infinity problem. The more sophisticated Link-State protocol used in OSPF and IS-IS provides a much better routing solution when the AS is large enough to have a legitimate hop count close to 16.

Provided that a RIP-1 implementation obeys the specification in RFC 1058, RIP-2 can interoperate with RIP-1. El formato de mensaje RIP se extiende como se muestra en figura siguiente.

Los campos en un mensaje RIP-2 son los mismos que los de RIP-1 excepto los siguientes:

Versión

Is 2. This tells RIP-1 routers to ignore the fields designated as ``must be zero'' (if the value is 1, RIP-1 routers are required to discard messages with non-zero values in these fields since the messages originate with a router claiming to be RIP-1-compliant but sending non-RIP-1 messages).

Familia de direcciones

May be X'FFFF' in the first entry only, indicating that this entry is an authentication entry.

Tipo de autentificación

Defines how the remaining 16 bytes are to be used. The only defined types are 0 indicating no authentication and 2 indicating that the field contains password data.

Datos de autentificación

The password is 16 bytes, plain text ASCII, left adjusted and padded with ASCII NULLs (X'00').

Etiqueta de ruta

Is a field intended for communicating information about the origin of the route information. It is intended for interoperation between RIP and other routing protocols. RIP-2 implementations must preserve this tag, but RIP-2 does not further specify how it is to be used.

Máscara de subred

The subnet mask associated with the subnet referred to by this entry.

Salto siguiente

A recommendation about the next hop that the router should use to send datagrams to the subnet or host given in this entry.

Para asegurar interoperación con RIP, el RFC 1723 especifica las restricciones siguientes para los routers RIP-2 que envían sobre una interfaz de red donde un router RIP-1 puede oir y operar sobre los mensajes RIP.

La información interna para una red nunca se debe advertir en otra red.

Information about a more specific subnet may not be advertised where RIP-1 routers would consider it a host route.

Supernet routes (routes with a subnet mask shorter than the natural or ``unsubnetted'' network mask) must not be advertised where they could be misinterpreted by RIP-1 routers.

RIP-2 also supports the use of multicasting rather than simple broadcasting. This can reduce the load on hosts which are not listening for RIP-2 messages. This option is configurable for each interface to ensure optimum use of RIP-2 facilities when a router connects mixed RIP-1/RIP-2 subnets to RIP-2-only subnets. Similarly, the use of authentication in mixed environments can be configured to suit local requirements.

RIP-2 is implemented in recent versions of the gated daemon, often termed gated Version 3. Since the draft standard is new at the time of writing, many implementations will comply with the earlier version described in RFC 1388. Such implementations will interoperate with those adhering to RFC 1723.

Para más información sobre RIP-2, ver:

RFC 1721 - RIP Versión 2 - Análisis del Protocolo

RFC 1722 - RIP Version 2 - Declaración de aplicabilidad del Protocolo

RFC 1723 - RIP Version 2 - Información Adicional de Acarreo

RFC 1724 - RIP Version 2 - Extensión MIB

Página 80: [3] Eliminado **Pablo Durazno** **11/02/2003 12:26:00**

Salto de página

Indice

Página 90: [4] Eliminado **Pablo Durazno** **12/02/2003 9:16:00**