University of Castilla-La Mancha

A publication of the
Department of Computer Science

Protocolos de Encaminamiento en Internet

by

Rafael Casado, Francisco J. Quiles, and José Duato

Technical Report **#DIAB-01-02-16** February 2001

DEPARTAMENTO DE INFORMÁTICA
ESCUELA POLITÉCNICA SUPERIOR
UNIVERSIDAD DE CASTILLA-LA MANCHA
Campus Universitario s/n
Albacete – 02071 – Spain
Phone +34.967.599200, Fax +34.967.599224

Protocolos de encaminamiento en Internet*

Rafael Casado, Francisco José Quiles
Departamento de Informática
Universidad de Castilla – La Mancha
Escuela Politécnica Superior
02071–Albacete, España
{ rcasado, paco}@info-ab.uclm.es

José Duato
Departamento de Informática de Sistemas y Computadores
Universidad Politécnica de Valencia
Facultad de Informática
46071–Valencia, España
jduato@gap.upv.es

1 El protocolo RIP

RIP (*Routing Information Protocol*) es un protocolo de encaminamiento que comenzó a utilizarse en ARPANET en 1969. En un principio, este protocolo funcionó bien en sistemas pequeños, pero resultó inadecuado a medida que los sistemas autónomos se volvieron más grandes. RIP padece de los problemas típicos de los algoritmos basados en vectores de distancia: cuenta a infinito y lenta convergencia. Por esta razón se reemplazó en mayo de 1979 por OSPF (*Open Shortest Path First*), un protocolo basado en la supervisión del estado de los enlaces, que se describe más adelante. La primera especificación de RIP [RFC1058] se escribió cuando ya existían múltiples implementaciones diferentes. No obstante, la interacción entre dichas implementaciones no supuso un problema infranqueable gracias a la simplicidad y robustez del encaminamiento basado en vectores de distancia.

Un vector de distancias es un conjunto de pares [destino, coste] que cada nodo transmite a sus vecinos. Tal y como se mencionó en el capítulo correspondiente, los algoritmos de encaminamiento basados en vectores de distancias limitan la distancia máxima a la que pueden encontrarse los destinos, con el objeto de paliar el problema de cuenta al infinito. Los implementadores de RIP, en una falta de previsión, asignaron este máximo a 16. Con esta cota superior, el único criterio de coste aplicable es el de número de saltos. La estructura del paquete utilizado por RIP se muestra en la Figura 1.1. El significado de cada campo es el siguiente:

*Este trabajo ha sido financiado por la CICYT española nº TIC97-0897-C04.

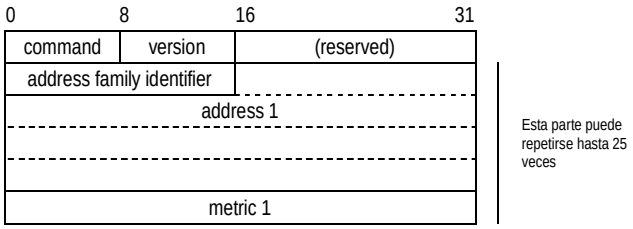

Figura 1.1. Formato de paquete RIP.

- **Command:** define el tipo de paquete, siendo 1 si es de petición (*request*), o 2 si es de respuesta (*response*). Un nodo envía un paquete de petición cuando se inicializa o cuando ha expirado cierta información acerca de un destino en particular. Si el paquete no contiene destinatarios concretos entonces se asume que se refiere a todos los destinatarios posibles. Un paquete de petición no contiene información sobre la distancia a la que se encuentran los destinos (el campo de métrica no está asignado), en cambio, un paquete de respuesta si contiene los costes asociados. Un nodo envía un paquete de respuesta ante la recepción de una petición, ante un cambio en el vector de distancias, o periódicamente (usualmente cada 30 segundos), pues los vectores de distancias expiran con el tiempo.
- **Version:** tiene un valor 1 para el protocolo RIP original, y un valor 2 para la versión RIP-2.
- **Address family identifier:** determina el esquema de direccionamiento de los nodos. RIP solo ha sido utilizado sobre IP con este campo asignado a 2.
- **Address:** El campo dirección tiene una extensión de 14 bytes para albergar el identificador del nodo destino. Si el esquema utilizado es IP, solo se utilizan 4 bytes (del tercero al sexto). No se contempla la posibilidad de incluir la máscara de la subred, a pesar de disponer de espacio para ello.
- **Metric:** contiene la distancia a la que se encuentra el destino. Tiene un tamaño de 32 bits, a pesar de que la máxima distancia (establecida a 16) requiere solamente 4 bits. La razón es que resulta conveniente asignar los campos importantes en direcciones múltiplo de 4 bytes.

Los tres últimos campos pueden repetirse un máximo de 25 veces. Es decir, un paquete puede contener un vector con 25 destinos, requiriéndose varios paquetes si el número de destinos es mayor.

2 El protocolo OSPF

El protocolo RIP basado en vectores de distancias fue reemplazado en 1979 por un protocolo basado en el estado de los enlaces. En 1988, la *Internet Engineering Task Force* (grupo de trabajo de ingeniería en Internet) comenzó a trabajar en un sucesor más sofisticado que llegaría a convertirse en estándar en 1990. Ese sucesor fue el protocolo OSPF (*Open Shortest Path First*, abrir primero la trayectoria más corta) [RFC 1131, RFC 1583]. Los requerimientos que OSPF debía reunir fueron los siguientes:

- Orientado a encaminar paquetes dentro de sistemas autónomos. Es, por tanto, un protocolo IRP (*Interior Router Protocol*).
- Publicación como literatura abierta (no propietaria), de ahí la "O" de *Open* en OSPF.
- Reconocimiento de una amplia variedad de métricas (con costes basados en diferentes tipos de distancias y retardos).
- Algoritmo dinámico que se adaptara fácil y rápidamente a los cambios.
- Clasificación del tráfico según su tipo de servicio asociado.
- Equilibrio de cargas.
- Reconocimiento de sistemas jerárquicos.
- Seguridad ante ataques intencionados.

En OSPF, cada enrutador construye y mantiene una base de datos que refleja la topología de la red, expresada en forma de grafo dirigido. La Figura 2.1 muestra un ejemplo de sistema autónomo. La Figura 2.2 muestra el grafo dirigido obtenido por OSPF para dicha red. Los nodos de dicho grafo son los enrutadores y las redes, a su vez divididas en *redes de tránsito* (si pueden transportar datos que ni se originan ni se reciben en la misma red), o redes *terminales*. Los enlaces pueden conectar dos enrutadores entre sí o un enrutador a una red. También es posible la conexión de un enrutador a un terminal. Se asocia un coste no nulo a los enlaces de salida de los enrutadores hacia las redes, y un coste nulo a los enlaces que parten desde la red hacia los enrutadores. Si un enrutador está conectado a un sistema autónomo distinto, se representa cada red del otro sistema como una red terminal conectada al enrutador a través de un enlace cuyo coste es determinado por un protocolo de encaminamiento exterior (ERP).

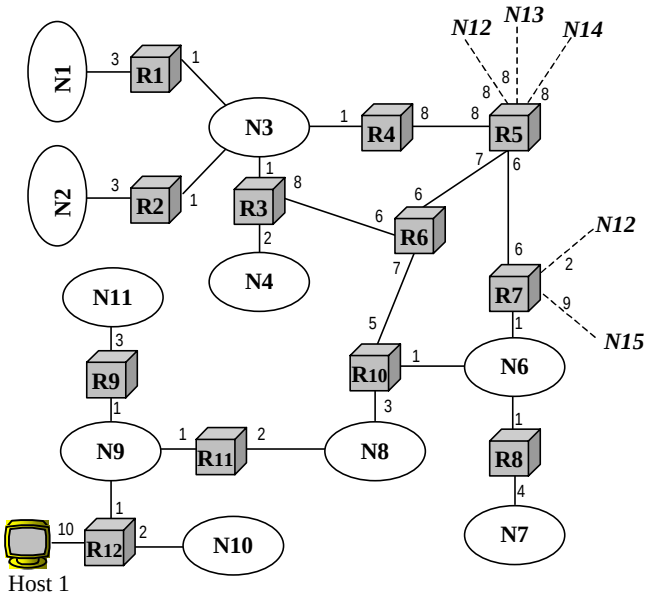

Figura 2.1. Ejemplo de sistema autónomo.

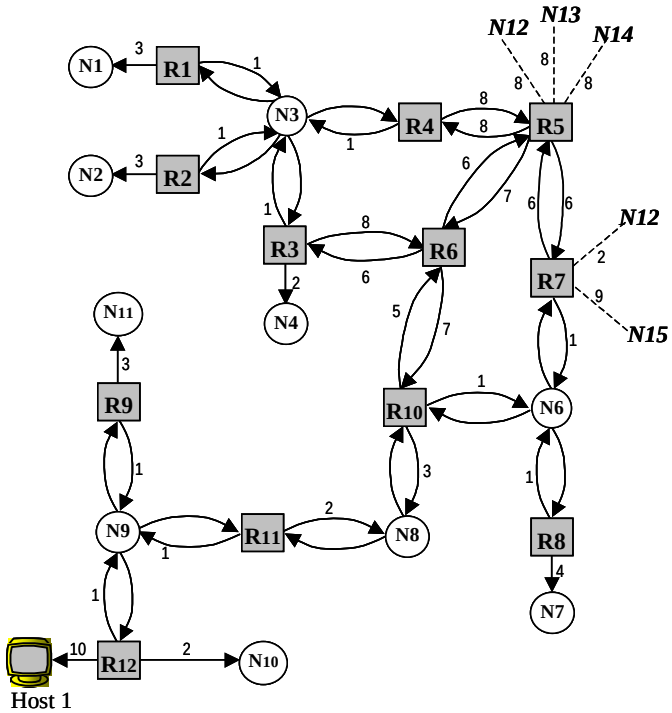

Figura 2.2. Grafo dirigido construido por OSPF para el SA de la .

A partir del grafo dirigido cada enrutador calcula su tabla de encaminamiento. Dicha tabla contiene rutas mínimas hacia todas las redes destino (sólo el siguiente salto para cada ruta, y no la ruta completa). Para esta tarea se aplica el algoritmo de Dijkstra. La Figura 2.3 muestra el árbol de expansión resultante de aplicar dicho algoritmo (con el nodo 6 como nodo fuente) al grafo de la Figura 2.2. La Tabla 2.1 muestra la tabla de encaminamiento de dicho nodo.

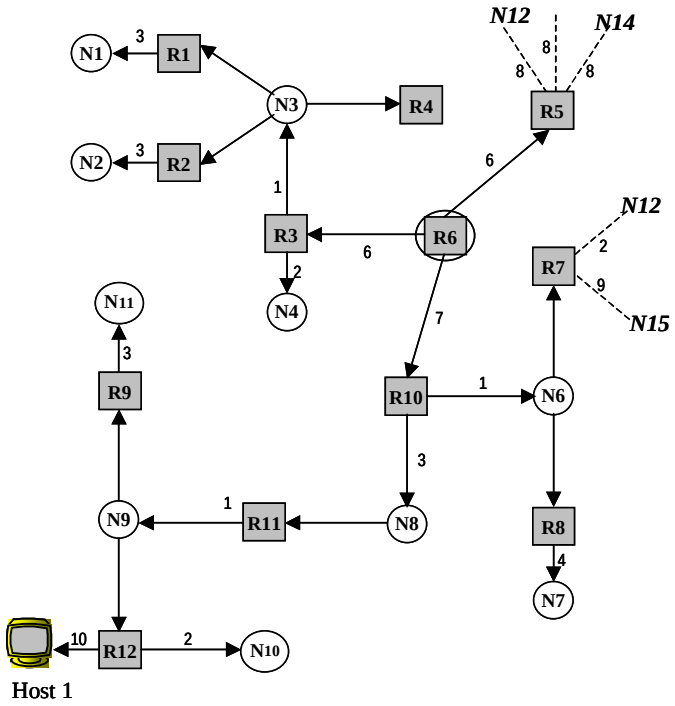

Figura 2.3. Árbol de expansión obtenido al aplicar el algoritmo de Dijkstra con *s* = 6.

Destino	Enrutador siguiente	Coste
N1	R3	10
N2	R3	10
N3	R3	7
N4	R3	8
N6	R10	8
N7	R10	12
N8	R10	10
N9	R10	11
N10	R10	13
N11	R10	14
N12	R10	10
N13	R5	14
N14	R5	14
N15	R10	17
H1	R10	21
RT5	R5	6

Tabla 2.1. Tabla de encaminamiento del enrutador 6.

2.1 Estructura jerárquica

Debido a que muchos AS son grandes y difíciles de manejar, OSPF permite su división en *áreas* numeradas, donde un área es una red o grupo de redes contiguas. Un área es una generalización de una subred y fuera de ella su topología no es visible. Cada AS tiene un área principal o

backbone, al que están conectadas el resto de las áreas en una distribución de estrella. OSPF clasifica los enrutadores en cuatro tipos, en función del área o áreas a las que pertenezcan:

- Enrutadores *internos*, contenidos en una única área.
- Enrutadores *de borde de área*, conectados a varias áreas. Estos enrutadores necesitan la base de datos de ambas áreas y deben realizar, para cada una por separado, la obtención de rutas óptimas.
- Enrutadores *de backbone*. Estos enrutadores aceptan información de los enrutadores de borde de área con el fin de calcular la mejor ruta a todos los enrutadores. Esta información se propaga de regreso a los enrutadores de borde de área, quienes la divulgan a su área. Usando esta información, un enrutador a punto de enviar un paquete inter-área puede seleccionar el mejor enrutador de salida al backbone. A continuación, el paquete atraviesa el backbone hasta alcanzar el enrutador de borde perteneciente al área de destino. Finalmente, el mensaje se desplaza desde el enrutador conectado al backbone hasta el nodo destino. Todos los enrutadores de borde de área son automáticamente parte del backbone.
- Enrutadores *de frontera de AS*, que se relacionan con enrutadores de otros AS.

Para asimilar cambios de topología, OSPF opera realizando intercambios de información entre enrutadores adyacentes. Es muy frecuente que estos estén conectados a través de una red de multidifusión (LAN). En estas situaciones, todos ellos consideran al resto como sus vecinos inmediatos, lo que conlleva que la sobrecarga añadida por el mecanismo de encaminamiento se incremente considerablemente. Estos efectos indeseables se evitan mediante la elección de un *enrutador designado*, el cual asume que sólo él es adyacente a todos los demás. El enrutador designado intercambia información con el resto de los enrutadores, y estos no intercambiarán información entre sí. Generalmente, tras un cambio de topología el enrutador designado es aquel que ya estaba designado antiguamente o (en caso de no haber ninguno) aquel con identificador de mayor prioridad.

2.2 Tipos de servicio

OSPF dispone de cinco tipos de servicio, los mismos que se utilizan en IPv4. El tipo de servicio se define en el campo TOS (*Type Of Service*) del paquete, y determina el significado del coste de los enlaces (o métrica). Veamos a continuación una breve descripción de cada tipo de servicio:

1. Normal (TOS 0): Es la métrica que se emplea por defecto, asignada por el administrador para satisfacer necesidades generales, y que es comprendida por todos los enrutadores.
2. Minimizar coste monetario (TOS 2): Métrica empleada si se puede asignar coste monetario al uso de la red.

3. Maximizar fiabilidad (TOS 4): Métrica basada en la historia reciente de fallos en la red o en tasas de paquetes erróneos.
4. Maximizar caudal (TOS 8): Esta métrica debe configurarse previamente basándose en la capacidad de cada enlace. Una magnitud muy utilizada es la duración de un bit en nanosegundos.
5. Minimizar retardo (TOS 16): Medida del retardo para un salto en particular, basada en el retardo de propagación y en el retardo en los buffers.

Para proporcionar estos cinco tipos de servicio, OSPF mantiene cinco grafos de topología distintos y sus correspondientes tablas de encaminamiento. Los datagramas IP suelen incorporan un campo TOS. Según el valor de este campo, cada enrutador consulta la tabla de encaminamiento apropiada para encaminar el datagrama. Si el datagrama no incluye el campo TOS entonces se usa la tabla correspondiente a la métrica por defecto (TOS 0).

2.3 Formato de los paquetes

OSPF es un protocolo que se ejecuta sobre IP, es decir, sus paquetes son transmitidos encapsulados dentro de paquetes IP, lo que se indica con el campo “protocolo” asignado a 89. Los paquetes OSPF tienen la misma cabecera de longitud fija, lo que favorece su codificación compacta y rápido procesamiento, a costa de reducir su extensibilidad futura. Esta cabecera se muestra en la Figura 2.4. El significado de cada campo es el siguiente:

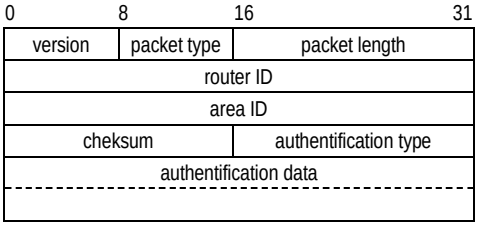

Figura 2.4. Cabecera común de los paquetes OSPF.

- **Version:** versión del protocolo (2 en la actualidad).
- **Packet type:** tipo de paquete (cada tipo se describe más adelante).
  1 = hello (saludo)
  2 = DD, database description (descripción de la base de datos)
  3 = link state request (solicitud de información)
  4 = link state update (actualización de información)
  5 = link state acknowledment (reconocimiento de actualización)
- **Packet length:** número de octetos del paquete.
- **Router ID:** dirección IP del enrutador emisor.

- **Area ID:** identificador del área a la que pertenece el paquete.
- **Checksum:** código de error (similar al utilizado en IP).
- **Authentification type:**
  0 = sin autentificación
  1 = con clave simple
  2-255 = actualmente indefinido
- **Authentification data:** clave de 64 bits

Veamos a continuación cada uno de los cinco tipos de paquetes OSPF. En primer lugar, los paquetes *hello* permiten detectar cambios en el estado de los vecinos o de los enlaces que unen a un nodo con sus vecinos. Siempre son transmitidos entre vecinos inmediatos, y nunca recorren más de un enlace. La Figura 2.5 muestra el formato de estos paquetes donde se ha obviado la cabecera. El significado de cada campo es el siguiente:

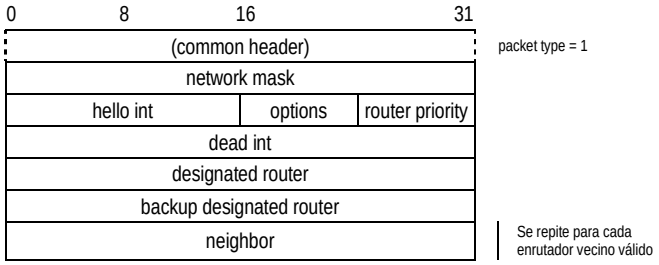

Figura 2.5. Paquete *hello* de OSPF.

- **Network mask:** la máscara configurada para este enlace en el enrutador emisor. Si el receptor no comparte este valor, entonces rechaza el paquete y no acepta al emisor como vecino.
- **Hello int:** intervalo entre emisión de paquetes *hello* (expresado en segundos). Este campo también debe coincidir con la información del receptor.
- **Options:** ciertas opciones como el soporte de múltiples métricas.
- **Router priority:** prioridad aplicada en la elección de enrutadores designados (principales y de reserva).
- **Dead int:** intervalo de tiempo (expresado en segundos) en que un enrutador considera a otro desactivado si no recibe paquetes *hello*.
- **Designated router:** identificador del enrutador que el emisor considera enrutador designado, o cero si no considera ninguno.
- **Backup designated router:** identificador del enrutador que el emisor considera enrutador designado de reserva, o cero si no considera ninguno.
- **Neighbors:** lista de identificadores de vecinos activos, es decir, aquellos de los que ha recibido paquetes *hello* dentro del intervalo delimitado en el campo *dead int*.

Cuando se activa un enlace entre dos enrutadores, estos deben sincronizar la información topológica que poseen. Para ello, los dos nodos aceptan una relación maestro/esclavo definida en función del UID. El maestro comunica su información topológica al esclavo mediante paquetes de descripción de la base de datos (DD, *database description*), utilizando tantos como sea necesario (cada paquete se identifica por un número de secuencia). Por su parte, el enrutador esclavo confirma cada paquete DD enviando paquetes DD al maestro con el mismo número de secuencia, pero conteniendo la propia información topológica. El maestro no envía un nuevo DD en tanto en anterior no haya sido confirmado. Cuando un nodo ha terminado de transmitir su información topológica continua emitiendo paquetes vacíos hasta que termine el otro. La Figura 2.6 muestra el formato de estos paquetes. El significado de cada campo es el siguiente:

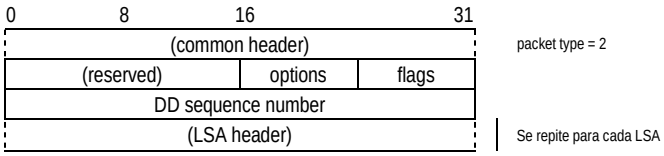

Figura 2.6. Paquete *database description*.

- **Options:** similar al campo *options* del paquete *hello*.
- **Flags:** bits cuyo estado activo indica que...
  MS (master/slave): el emisor es el nodo maestro
  M (more): no es el último paquete DD
  I (init): es el primer paquete DD
- **DD packet sequence number:** número orden en la secuencia de paquetes de descripción de topología.
- **LSA header:** es la parte común a diferentes tipos de LSA (*link state advertisement*). Puede repetirse varias veces dentro del paquete. Como otros paquetes también incorporan LSAs, estos se describen al final.

En cualquier momento, un enrutador puede solicitar información topológica a otro enrutador vecino. Para ello utiliza un mensaje *link state request*. La Figura 2.7 muestra el formato de estos paquetes. Los campos *LS type*, *link state ID* y *advertising router* son en realidad un fragmento de la cabecera de un LSA y se describen más adelante, en la descripción completa del LSA.

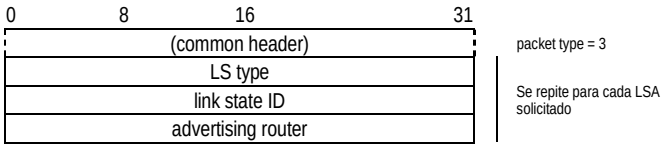

Figura 2.7. Paquete *link state request.*

Los paquetes *Link state update* contienen uno o varios LSAs. La Figura 2.8 muestra el formato de estos paquetes.

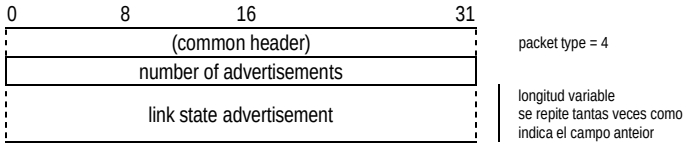

Figura 2.8. Paquete *link state update*.

Los paquetes *Link state acknowlegment* son paquetes de reconocimiento que proporcionana su confiabilidad a OSPF. Cada uno puede reconocer varios LSAs. La Figura 2.9 muestra el formato de estos paquetes.

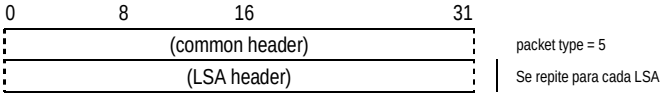

Figura 2.9. Paquete *link state acknowlegment*.

2.4 Información contenida en un LSA

Existen cinco tipos de LSAs. Todos ellos tienen una cabecera común. Primero se describe esta cabecera y después se comenta la información exclusiva de cada tipo. La cabecera tiene una longitud fija de 20 bytes, cuyo formato se muestra en la Figura 2.10. El contenido de cada campo es el siguiente:

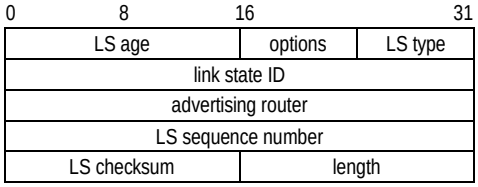

Figura 2.10. Cabecera de un LSA.

- **LS age:** edad estimada de la información (expresada en segundos).
- **LS type:** los valores posibles son los siguientes...
  1 = enlaces del enrutador
  2 = enlaces de la red
  3 = resumen de enlaces (subredes IP alcanzables)
  4 = resumen de enlaces (enrutadores alcanzables de sistemas vecinos)
  5 = resumen de enlaces (subredes IP alcanzables de sistemas vecinos)
- **Link state ID:** su significado depende del valor del campo anterior:
  Type = 1 → el ID del enrutador que generó la información
  Type = 2 → la dirección IP del enrutador designado de la LAN

Type = 3 → la dirección IP del enlace que conecta la subred
Type = 4 → el ID del enrutador de borde
Type = 5 → la dirección IP del enlace que conecta la subred

- **Advertising router:** ID del enrutador que generó la información.
- **LS sequence number:** número de secuencia del LSA.
- **LS checksum:** código de redundancia ISO 8473 (anexo C).
- **Length:** tamaño en bytes.

Veamos a continuación las diferencias entre los distintos tipos de LSAs. El primer tipo se denomina *router links advertisement,* cuyo formato se muestra en la Figura 2.11. El contenido de cada campo es el siguiente:

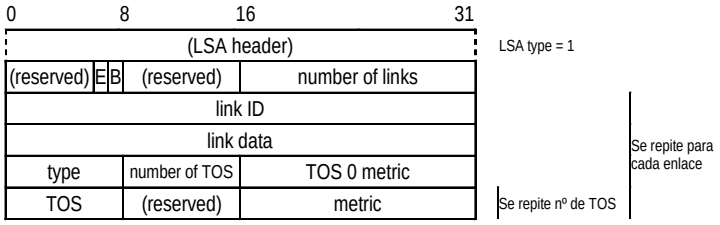

Figura 2.11. LSA *router links advertisement*.

- **Bit E (external):** indica que el emisor es un enrutador frontera de AS.
- **Bit B:** indica que el emisor es un enrutador de borde de área.
- **Number of links:** número total de enlaces que el enrutador emisor tiene en el área.
- **Link ID:** identifica al componente conectado al otro extremo del enlace. Para cada tipo de enlace (ver campo type más adelante), este identificador representa...
  Type = 1 → el ID de un enrutador vecino
  Type = 2 → la dirección IP del enrutador designado de una LAN
  Type = 3 → el número de una subred IP
  Type = 4 → el ID de un enrutador vecino
- **Link data:** para enlaces tipo 3 contiene la máscara de subred. En otro caso contiene la dirección IP del enrutador que generó el LSA sobre este enlace.
- **Type:** tipo de enlace:
  1 = enlace punto a punto con otro enrutador
  2 = conexión a una LAN de tránsito
  3 = conexión a una LAN final
  4 = enlace virtual
- **Number of TOS:** especifica el número de métricas incluidas para este enlace, sin contar TOS 0 que es obligatoria.
- **TOS:** tipo de servicio. Los tipos disponibles en OSPF son similares a los proporcionados por IP en este mismo campo.

El segundo tipo de LSA soportado por OSPF se denomina *network links advertisement*. Su formato se muestra en la Figura 2.12. El contenido de cada campo es el siguiente:

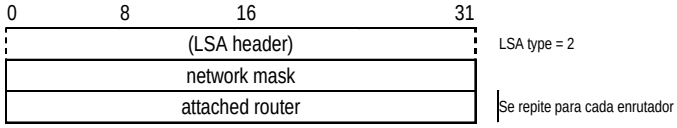

Figura 2.12. LSA *network links advertisement*.

- **Network mask:** es la máscara IP de la LAN.
- **Attached router:** es el identificador del enrutador de la LAN que es vecino del enrutador designado.

Los tipos de LSA 3 y 4 se denominan *sumary link advertisement*. Estos LSAs son generados por los enrutadores de borde de área y transmitidos a toda el área por inundación. Un LSA de tipo 3 notifica el coste de alcanzar subredes situadas fuera del área. Un LSA de tipo 4 notifica el coste de alcanzar un enrutador de frontera de AS. El formato de estos LSAs se muestra en la Figura 2.13. El contenido de cada campo es el siguiente:

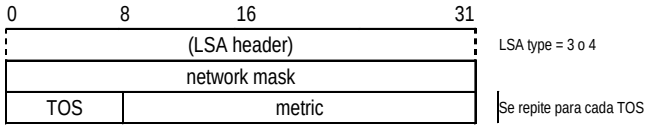

Figura 2.13. LSA *sumary link advertisement*.

- **Network mask:** sólo es relevante para LSAs de tipo 3. Los LSAs de tipo 4 tienen este campo asignado a 0.

El quinto y último tipo de LSAs se denomina *AS external link advertisement*. Estos LSAs son generados por enrutadores de frontera de AS y transmitidos por inundación al resto del AS (excepto a áreas finales). El formato se muestra en la Figura 2.14, y el contenido de cada campo es el siguiente:

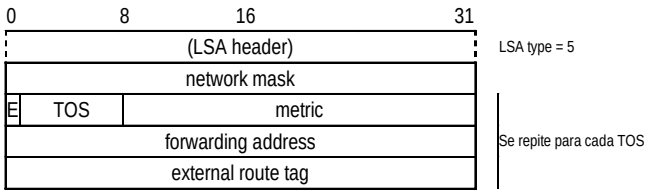

Figura 2.14. LSA *AS external link advertisement*.

- **Network mask:** máscara del nodo destinatario notificado.
- **Bit E:** un nivel bajo indica que la métrica utilizada puede ser considerada similar a la utilizada para ese tipo de servicio dentro del AS.
- **TOS:** tipo se de servicio. Similar al campo TOS de IP pero sin bits de precedencia.

- **Metric:** coste para alcanzar al destino.
- **Forwarding address:** la información de este campo permite optimizar el último salto realizado por los paquetes. Si el AS notificador advierte que el destino puede ser alcanzado por un camino más corto siguiendo otro enrutador de la misma LAN, entonces el AS notificador pone la dirección IP de ese enrutador en este campo.
- **External route tag:** espacio reservado para información adicional suministrada por protocolos interdominio.

3 El protocolo BGP

El protocolo BGP (*Border Gateway Protocol*) es el protocolo de encaminamiento interdominio más utilizado en Internet. Una descripción rigurosa de la versión actual (BGP-4) puede encontrarse en [RFC 1771]. Este protocolo se diseñó para permitir la cooperación en el intercambio de información de encaminamiento entre enrutadores de distintos sistemas autónomos. En terminología BGP los enrutadores se denominan pasarelas (*gateways*), y realizan tres procesos funcionales: adquisición de vecinos, detección de vecinos alcanzables y detección de redes alcanzables.

El termino *adquisición de vecinos* implica que dos dispositivos de encaminamiento que comparten la misma subred física, pero que pertenecen a distintos sistemas autónomos, deciden intercambiar regularmente información de encaminamiento. Se requiere un procedimiento formal para la adquisición ya que uno de los dos dispositivos puede decidir no participar. En este procedimiento un dispositivo hace una oferta a otro (mediante un mensaje *open*), el cual puede aceptarla (mediante un mensaje *keepalive*) o rechazarla.

Una vez establecida la relación de vecindad, se utiliza el procedimiento de *detección de vecino alcanzable* para mantenerla. Cada miembro necesita estar seguro de que su pareja existe y está todavía comprometida con la relación. Para este propósito, periódicamente ambos dispositivos de encaminamiento se envían mensajes *keepalive*.

El último procedimiento es la *detección de redes alcanzables*. Cada pasarela mantiene una base de datos con las subredes que puede alcanzar y la ruta completa para hacerlo. Siempre que se modifica esta base de datos, la pasarela lo notifica a todos los demás dispositivos de encaminamiento que implementan BGP, por medio de paquetes *update*. De esta forma, el resto de pasarelas puede actualizar su propia información.

3.1 Formato de los paquetes

A diferencia de OSPF, BGP es un protocolo situado sobre el nivel de transporte, es decir, los paquetes BGP son transmitidos encapsulados dentro de paquetes TCP. Esto permite asumir que el intercambio de información se realiza de forma confiable. Los cuatro tipos de paquetes BGP tienen la cabecera mostrada en la Figura 3.1. El contenido de cada campo es el siguiente:

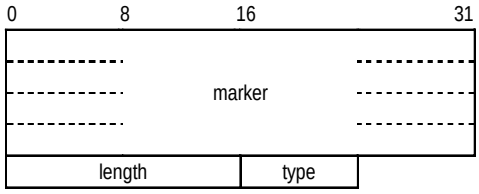

Figura 3.1. Cabecera de un mensaje BGP.

- **Marker:** el emisor puede insertar un valor de hasta 16 bytes en este campo. Este valor sería usado como parte de un mecanismo de autentificación que permita al destinatario verificar la identidad del emisor. En realidad, este campo se reserva (tanto en BGP como en otros protocolos que incorporan un campo similar) para el día en que se desarrolle un esquema de autentificación eficaz. Actualmente tiene todos sus bits asignados a uno.
- **Length:** longitud en bytes del paquete, incluyendo la cabecera.
- **Type:** tipo de paquete. Existen cuatro posibilidades:
  1 = open
  2 = update
  3 = notification
  4 = keepalive

Para adquirir un vecino, un dispositivo de encaminamiento establece primero una conexión TCP. Para ello se utiliza un paquete *open* mostrado en la Figura 3.2. Este mensaje identifica al AS al que pertenece el emisor y suministra la dirección IP del dispositivo de encaminamiento. La descripción de cada campo es la siguiente:

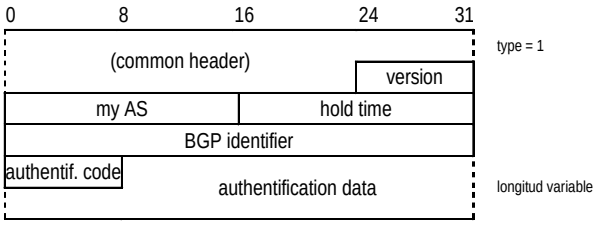

Figura 3.2. Formato de paquete *open*.

- **Version:** versión del protocolo, actualmente la 4.
- **My autonomous system:** indica el numero de AS del emisor.

- **Hold time:** tiempo que tiene que esperar el receptor antes de asumir que el emisor a caído. El emisor debe continuar emitiendo paquetes antes de que este tiempo se agote.
- **BGP identifier:** dirección IP del emisor. BGP considera como identificador de cada enrutador su propia dirección IP.
- **Authentification code:** define el sistema de autentificación empleado. En la actualidad este campo se asigna a cero.
- **Authentification data:** datos destinados a la autentificación del paquete. La longitud y el contenido de este campo dependen del campo anterior. De momento este campo no se utiliza y tiene una longitud de cero bytes.

Un mensaje *update* facilita dos tipos de información:

1) Información sobre una ruta particular a través del conjunto de redes. Dicha ruta se incorpora a la base de datos de cada dispositivo de encaminamiento que la recibe.
2) Una lista de rutas que fueron previamente anunciadas por este dispositivo de encaminamiento, y que ahora han sido eliminadas.

Estos dos tipos de información pueden ser proporcionadas simultáneamente en un único paquete. El formato del paquete *update* se muestra en la Figura 3.3. El contenido de cada campo se describe a continuación:

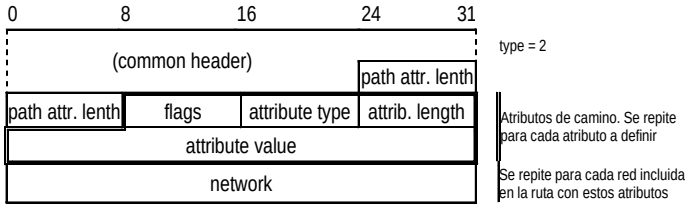

Figura 3.3. Formato de paquete *update*.

- **Path attributes length:** longitud de rutas no factibles. Número de atributos de la ruta.
- **Flags:** diversos bits que indican la opcionalidad, transitividad y parcialidad del atributo.
- **Attribute type:** existen cinco tipos de atributos:

  1 = *origin*; el atributo ocupa 1 byte e indica si la información fue generada por un protocolo de encaminamiento interior (como OSPF) o por un protocolo de encaminamiento exterior (en particular, BGP).

  2 = *AS path*; el atributo es de longitud variable y enumera una lista de AS que atraviesa la ruta.

  3 = *next hop*; el atributo ocupa 4 bytes y proporciona la dirección IP del dispositivo de encaminamiento frontera que se debe usar para alcanzar los destinos indicados en el campo *network*.

  4 = *unreachable*; el atributo no ocupa lugar adicional

  5 = *inter-AS metric*; el atributo ocupa 2 bytes

El tercer tipo de paquete es el paquete *notification*. Un paquete de notificación es enviado por el enrutador R1 al enrutador R2 para explicar por qué deniega la conexión a R2. Su formato se muestra en la Figura 3.4. El contenido de cada campo es el siguiente:

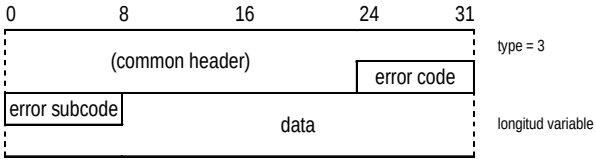

Figura 3.4. Formato de paquete *notification*.

- **Error code:** para cada código de error existen diversos subcódigos que no detallaremos:
  1 = cabecera corrupta. El tipo de paquete es inaceptable, bien por errores de sintaxis o por cuestiones de autentificación.
  2 = paquete *open* corrupto
  3 = paquete *update* corrupto
  4 = tiempo de *hold down* expirado
  5 = Error en la máquina de estados finitos (errores de procedimiento)
  6 = Cese (cierre voluntario de una conexión)
- **Data:** contiene el comando ofensivo.

El cuarto y último tipo de paquete es el paquete *keepalive*. Este paquete no tiene otra información aparte de la cabecera mostrada en la Figura 2.10. Este paquete se envía para inicializar el temporizador *hold down* del receptor antes de que expire, y el emisor no tiene información de interés que comunicar.