# Redes de Datos 2

## Enrutamiento Interno

# Temario: Enrutamiento interno

- Generalidades.
- Protocolos basados en vector distancia
- Protocolos RIP y  RIPng (RIP para IPv6)
- Protocolos basados en estado de enlace
- Protocolo OSPF
- OSPF para IPv6
- Protocolo ISIS

# Capa de red

- Objetivo principal:
  - Hacer llegar los paquetes desde A hasta B
- Identificación y localización de equipos: direcciones
- Dos planos:
  - Plano de Control:
    - Determina los (mejores) caminos a seguir por los paquetes
    - Función de enrutamiento (routing)
  - Plano de Datos:
    - Reenvía cada paquete recibido utilizando los caminos calculados por el plano de control
    - Función de encaminamiento (forwarding)

# Plano de datos: encaminamiento (forwarding)

- Cada nodo recibe un paquete por una de sus líneas de entrada o interfaces, decide en base a una tabla cuál es la línea de salida más adecuada para ese destino y encamina el paquete hacia allí
- Esas tablas se llaman tablas de forwarding
- En el caso tradicional se utiliza solamente la dirección de destino para elegir el próximo salto
- Escala de tiempo: nanosegundos
- En IP se utiliza el algoritmo de “longest prefix match”
  - Tanto en IPv4 como en IPv6
- Se encuentra distribuido

# Plano de Control: ruteo (routing)

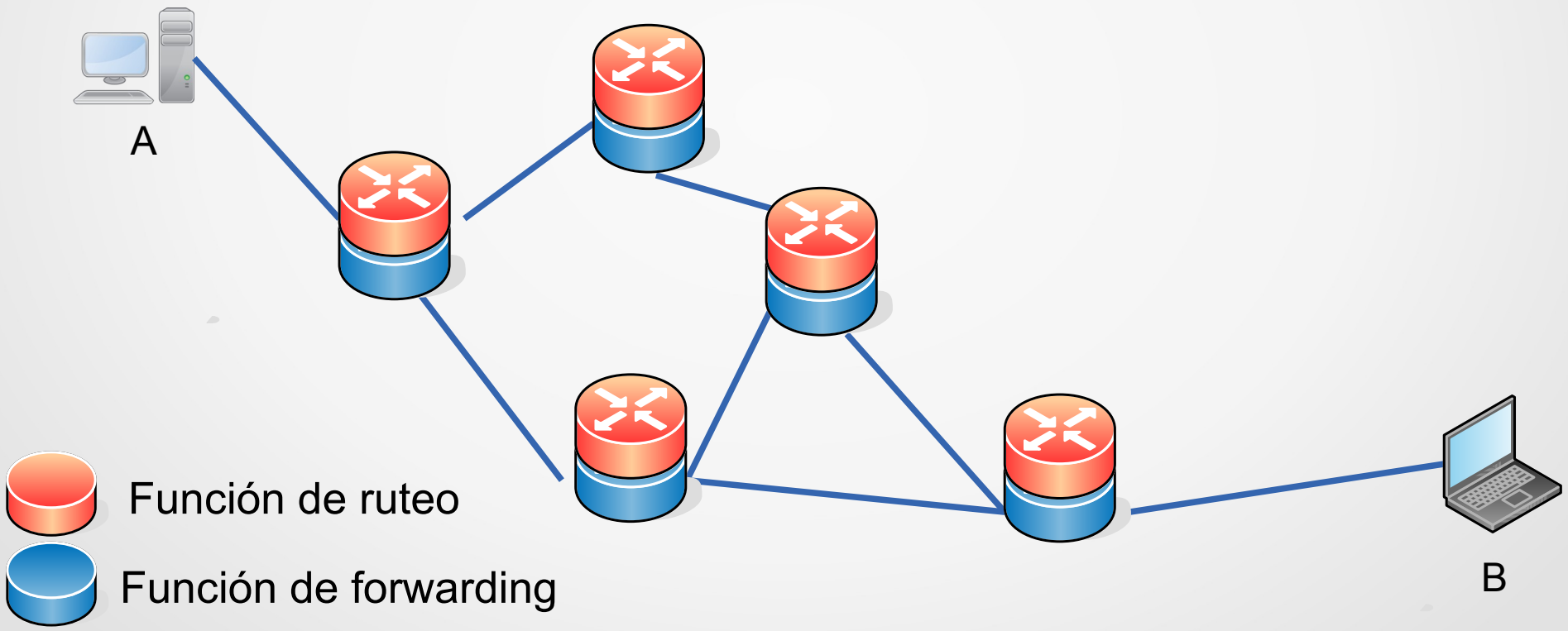
- Encontrar el mejor camino para ir de A a B
- El enrutamiento (routing) comprende las decisiones que determinan los caminos que deben seguir los paquetes desde un origen hacia un destino
  - En base a esto arma las tablas de forwarding
- Escala de tiempo: segundos
- Puede ser centralizado o distribuido
- Puede ser estático (configurado) o dinámico
- Puede convivir más de un mecanismo
- En caso dinámico, son aplicaciones en capa de aplicación

# Protocolos y algoritmos

- Para determinar el estado de la red y obtener la información de otros equipos, debemos intercambiar información: Protocolos de ruteo
- Para decidir el mejor camino una vez que tengo la información: algoritmos de ruteo
- Los algoritmos pueden ejecutarse de forma centralizada (“SDN”) o distribuida (tradicional)

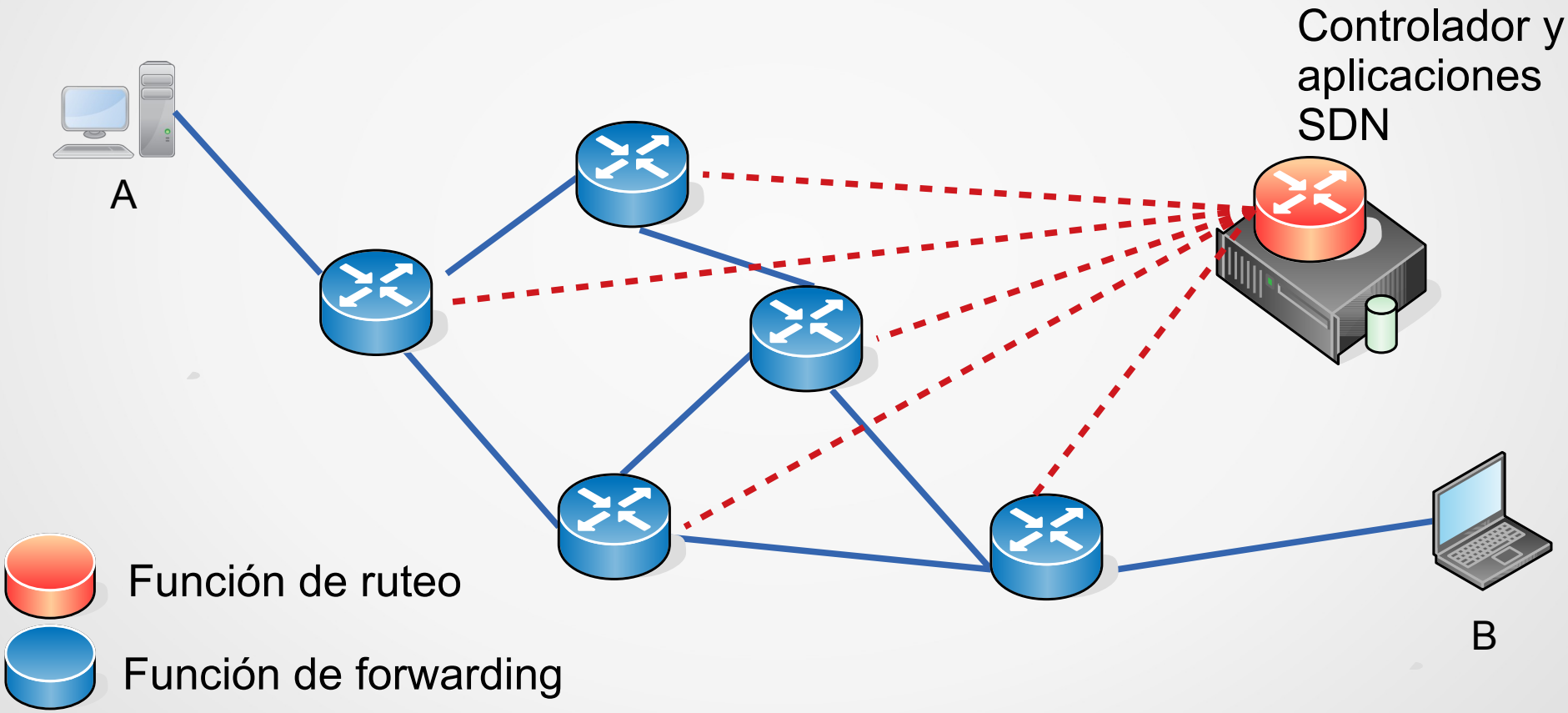
# Arquitectura tradicional

- Ruteo y forwarding integrados en cada equipo



# Arquitectura Software Defined Network

- Arquitectura SDN: enrutamiento centralizado



# Enrutamiento interno y externo

- Enrutamiento interno: dentro de un sistema autónomo
  - Objetivos:
    - eficiencia
    - elección “técnica” del mejor camino
    - Rápida convergencia
- Enrutamiento externo: entre sistemas autónomos
  - Objetivos:
    - Escalabilidad
    - Flexibilidad para realizar políticas
    - Interoperabilidad

# Clasificación algoritmos enrutamiento dinámico

- Centralizados:
  - Se conoce el grafo de la red con sus nodos, enlaces y costos
  - El cálculo puede ser centralizado (SDN) o distribuído (tradicional)
  - Estos algoritmos se conocen como de “estado del enlace” o “link-state (LS)” ya que conocen el costo (o estado) de todos los enlaces de la red
- Descentralizados
  - Los caminos se calculan de forma iterativa y distribuida en los enrutadores
  - No hay un nodo que tenga la información completa de todos los enlaces
  - Se conocen como de “vector distancia” o “distance-vector (DV)” ya que cada enrutador mantiene e intercambia con los demás una tabla de distancias aprendidas a los destinos de la red

# Principio de optimalidad

- Se usa el principio de optimalidad para determinar el “mejor camino”
- Si el nodo J está en el camino óptimo entre I y K, entonces el camino óptimo entre J y K está en la misma ruta
- Implica que se puede construir un árbol para cada destino (sink tree) con los caminos óptimos
- En condiciones estáticas me asegura no tener loops
  - Puede aparecer problema de loops transitorios por cambios de topología

# Distancias en protocolos de enrutamiento

- Precisamos criterios para poder definir cuándo un camino es mejor que otro
  - Cantidad de saltos
  - Capacidad de los enlaces
  - Retardo de propagación en los enlaces
  - Utilización del enlace
- El o los criterios elegidos por el protocolo se resumen usualmente en un *número* que llamaremos **distancia** o **costo**
- Intentaremos minimizar el “costo” de llegar a cada destino
- Los protocolos de enrutamiento interno en uso utilizan distancias sencillas ya sea administrativas o relacionadas con la cantidad de saltos o la capacidad de los enlaces

# Algoritmo de Vector distancia

- Algoritmo distribuido, iterativo, asíncrono
- Se mantiene una tabla con la distancia a todos los destinos conocidos
  - Destinos: redes
- Inicialmente el enrutador conoce solamente los destinos directamente conectados
- Periódicamente se envía a los enrutadores vecinos la tabla de todos los destinos conocidos y sus distancias
- En base a la información recibida de los vecinos, actualiza su tabla

# Algoritmo de vector distancia (cont.)

- Basado en la ecuación de Bellman-Fort:

  $d_x(y) = \min_j\{ c(x,j) + d_j(y) \}$

  - $d_x(y)$ distancia a y desde el nodo x
  - j son los vecinos directamente conectados a x
  - c(x,j) distancia desde x a j (directamente conectado)
  - La distancia de x a y pasando por j resulta de sumar la distancia de x a j más la distancia de j a y
  - El nodo j* que minimice la distancia, será el siguiente nodo en el mejor camino hacia y

# Funcionamiento de vector distancia

- Cada nodo recibe información de sus vecinos: tabla (lista) de distancias a todos los destinos conocidos (“vector de distancias”)
- Aplica Bellman-Fort:
  - Calcula las distancias a los distintos destinos pasando por cada vecino
  - Elige la menor distancia a cada destino
  - Construye su propio vector (tabla)
- El vector  así calculado es enviado a los vecinos
- Esto se repite periódicamente o cuando hay cambios

# Protocolo basado en vector distancia: Routing Information Protocol (RIP)

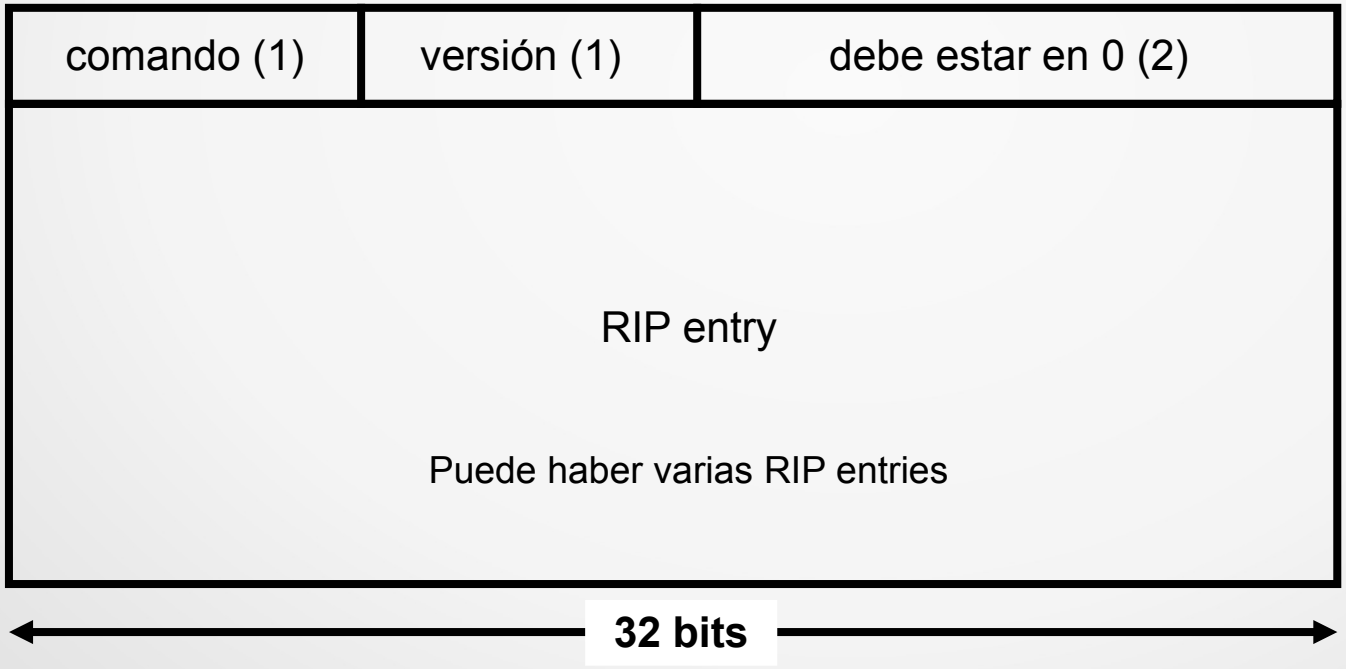
- ~~RIP v1 (RFC 1058): obsoleto~~
- RIP v2 (RFC 2453, STD 56): IPv4
- RIPng (RFC 2080 ): IPv6
  - RIPng es muy similar a RIP v2

# RIP

- Adecuado para redes pequeñas, con pocos o ningún camino alternativo
- Distancia: cantidad de saltos
- Problema de conteo a infinito en todas las versiones
- Radio máximo: 15 saltos

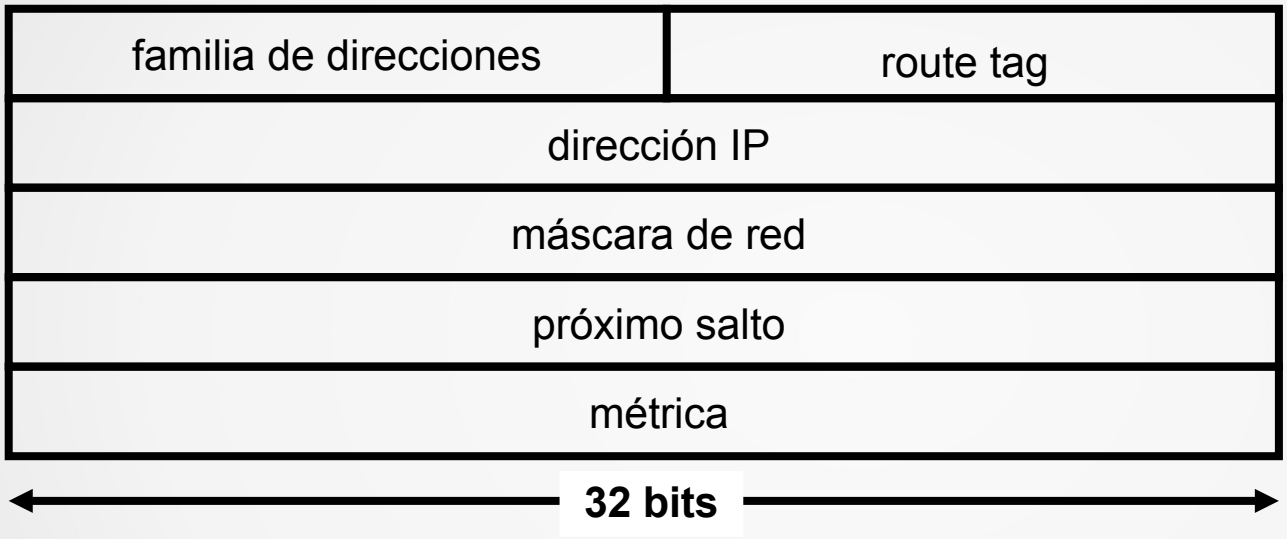
# Mensajes RIP

- El formato de los mensajes RIP es igual en todas las versiones:

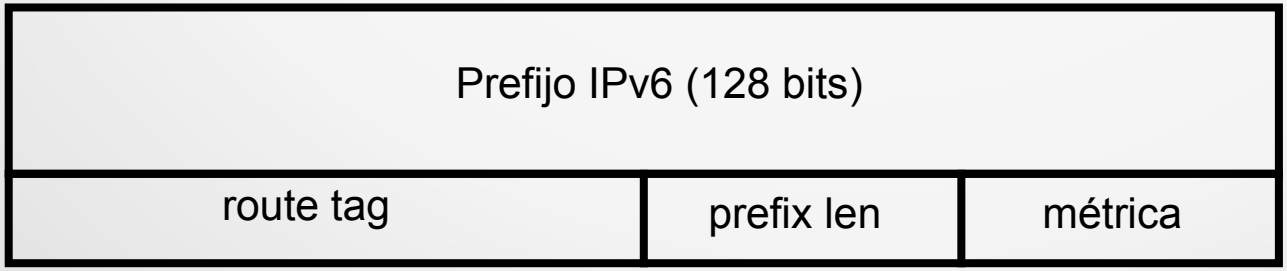


# RIP entries

- Para RIPv2




- Para RIPng



# RIP entries especiales

- En IPv4, RIP entry especial para autenticación
- En RIPng, RIP entry para indicar el Next Hop
  - Se identifica por métrica 255
  - El next-hop debe ser de tipo link-local
  - 0::0 significa “IP de quien envía el mensaje”
  - No es obligatorio tomarlo en cuenta

# Envío de información

- Se envía periódicamente (30 s) toda la información de prefijos y distancias
  - Antes si hay cambios
- Se envía a una dirección de multicast
  - FF02::9 para RIPng, 224.0.0.9 para RIPv2
- Puerto UDP 520 para RIPv2, UDP 521 para RIPng

# Repaso multidifusión (Multicast)

- Capa 2: en redes multiacceso como Ethernet (802.3) o WiFi (802.11), trama destinada a múltiples estaciones
  - Rango de direcciones de capa MAC reservado para esto
- Capa 3: Rango de direcciones IPv4/IPv6 reservado para comunicaciones 1 a muchos
  - 224.0.0.0/4 - FF00::/8
- “Grupo de multicast”: Una dirección de multicast
- Las estaciones pueden decidir “escuchar” grupos de múlticast
- IGMP en IPv4, ICMP en IPv6, solicitud para unirse a un grupo
- Algunas direcciones reservadas para usos particulares

# Uso de RIP

- Muy poco usado hoy en día
- Redes sencillas (topología sencilla, con mínimos loops)
- Radio pequeño
- Ventajas: simplicidad, mínimo uso de CPU y memoria

# Algoritmo de Dijkstra

- Dado un grafo con sus costos asociados, nos permite encontrar los mejores caminos desde un origen a todos los destinos
- Requiere la información de adyacencias (quién es vecino con quién) y costos asociados

# Protocolos basados en estado de enlace

- Idea: Construir el grafo de la red
- A partir del grafo, calcular mejores rutas a todos los destinos
- Paso 1: Descubrimiento de topología local
- Paso 2: Inundación confiable de la información a toda la red
- Paso 3: Construcción del “grafo” de la red y cálculo de los caminos más cortos usando Dijkstra
- Paso 4: Construcción de  la tabla de rutas

# Las piezas del “rompecabezas”

- Para cada nodo que pertenece al grafo (vértice) precisamos mínimamente:
  - Tipo de nodo
  - Id: Identificador unívoco de ese nodo
  - Lista de las aristas que lo unen con otros nodos
    - Para cada conexión precisamos el tipo de conexión, el costo de utilizarla, y el Id del nodo al cual se conecta
- Teniendo todos los nodos podemos listar todas las adyacencias y utilizar Dijkstra para encontrar los mejores caminos

# Características generales

- Convergencia rápida
- Consumo de memoria (guarda información de toda la topología)
- Consumo de CPU (cálculo de los mejores caminos usando Dijkstra cada vez que hay un cambio)
- CPU y memoria no son un problema hoy en día
- Se optimizan dividiendo la red en zonas y se realiza enrutamiento jerárquico
- Usualmente menos tráfico ya que mayormente intercambian diferencias con el estado anterior

# OSPF: Open Shortest Path First

- Protocolo de enrutamiento interior
- Estado de enlace
- RFC 2328 (STD 54) 1998
- Versiones anteriores: RFC 2178 ,RFC 1583, RFC 1247
- Jerárquico
- Soporta balanceo de tráfico por múltiples caminos de igual peso

# Elementos de la base de datos de estado de enlace (OSPF con 1 área)

- La información de estado de enlace se almacena y anuncia mediante lo que se conocen como LSA: Link State Advertisement (anuncios de estado de enlace)
- Dentro de un área, alcanza con la información de enrutadores y redes multiacceso (2 tipos de LSA)
  - Router: estado de las interfaces y adyacencias
  - Red: IP/máscara y enrutadores conectados
- Al agregar áreas e información externa aparecen nuevos LSA
- Cada enrutador guarda una lista de todos los LSA recibidos (base de datos de estado de enlace)

# Descubrimiento de topología local

- El enrutador conoce (por configuración) las interfaces a las que está conectado y las direcciones correspondientes
- Descubre (protocolo Hello) a sus vecinos
- Tiene la métrica de utilizar cada interfaz
- Sabe a qué redes llega directamente
- Con esta información, genera la información que lo describe

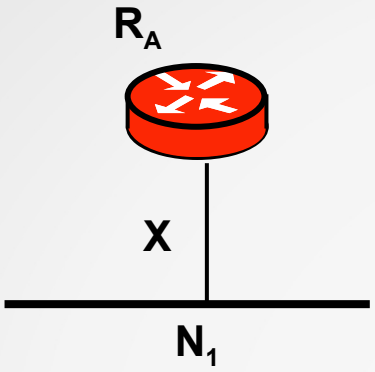
# ¿Qué informa cada enrutador sobre si mismo?

- Identificación del enrutador (router-id)
  - Identifica unívocamente al enrutador
  - Número de 32 bits
  - Si no se configura el enrutador puede elegir una dirección IP, usualmente mayor IP de las interfaces de loopback (NO 127.0.0.1), de lo contrario, mayor IP del enrutador
- Para todos los enlaces (interfaces):
  - Tipo de enlace
  - El costo (métrica) de usar la interfaz
  - A qué está conectado el enlace (id del nodo en el otro extremo (un enrutador o una red de tránsito), o una red stub)
  - Otros parámetros

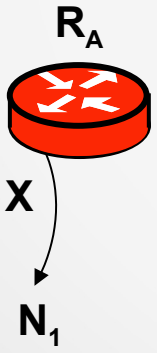
# Tipos de tecnologías de capa de enlace para OSPF

- Acceso múltiple con broadcast
  - Ej. Ethernet
- Punto a Punto
  - Ejemplo: línea serial, PVC ATM o Frame Relay
- NBMA (acceso múltiple sin broadcast)
  - Ej.: ATM, Frame Relay (mallados). Fuera de uso.
  - Requiere configuración manual de vecinos
- Punto – Multipunto
  - Conectividad “Uno a muchos”. Muy poco uso
  - En el grafo se representan como múltiples enlaces punto a punto
- Links virtuales

# Tipos de enlace para OSPF: Redes “stub”




Grafo correspondiente



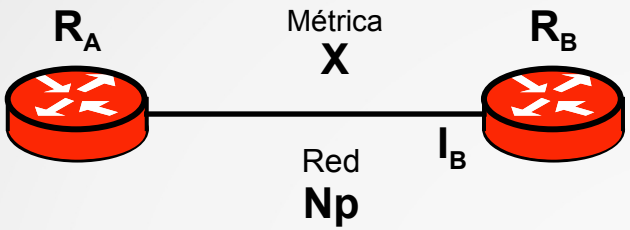

**X**: valor de métrica
**N1**: red conectada (IP/máscara)

Tabla de adyacencias

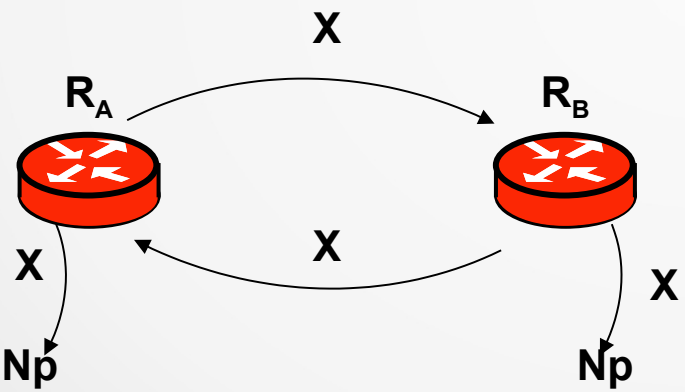
| Hacia \ Desde | $R_A$ | $N_1$ |
|---|---|---|
| $R_A$ | | |
| $N_1$ | X | |

# Tipos de enlaces para OSPF: Punto a punto

Enlace Punto a Punto




Grafo correspondiente



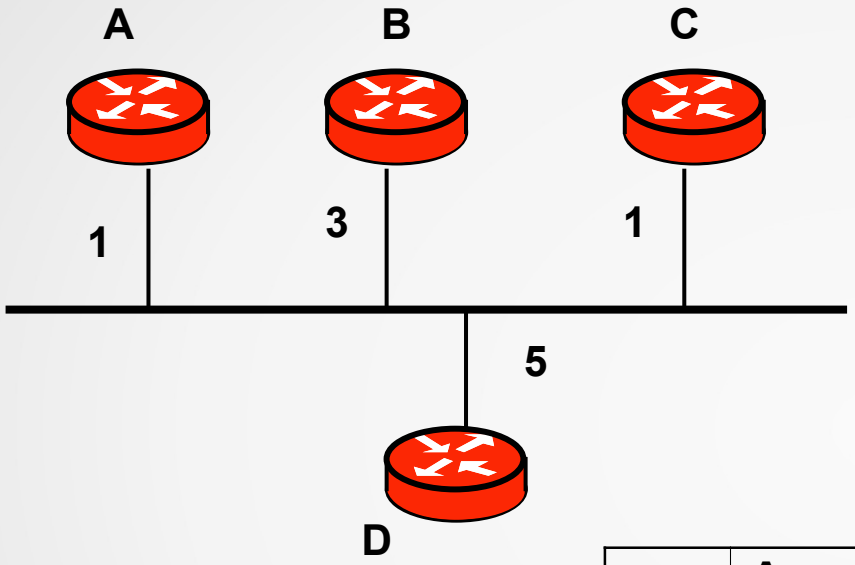

Hacia / Desde

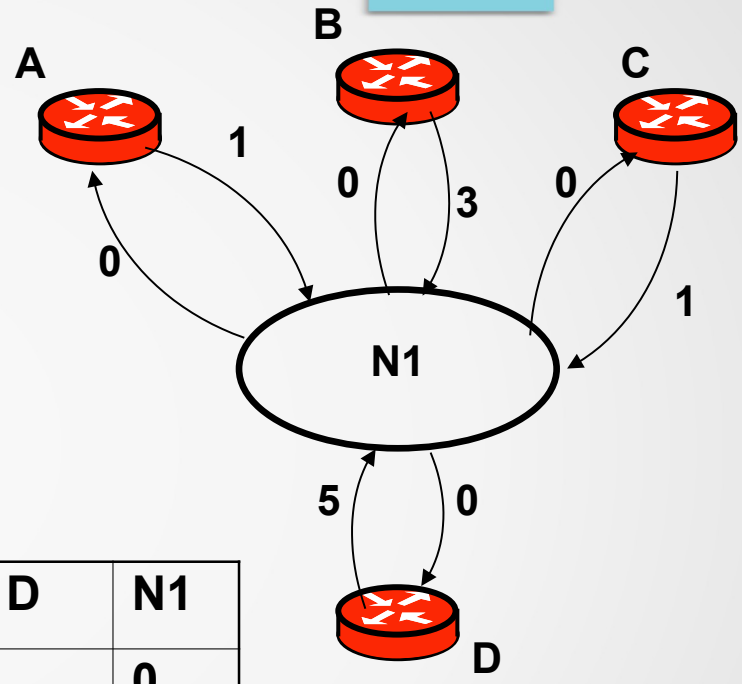
| | $R_A$ | $R_B$ |
|---|---|---|
| $R_A$ | | X |
| $R_B$ | X | |
| Np | X | X |

**La red Np se representa en el LSA como una red STUB conectada a ambos enrutadores**

**Las interfaces pueden no estar numeradas, en cuyo caso no tendremos dicha red STUB**

# Redes multiacceso (broadcast o NBMA)

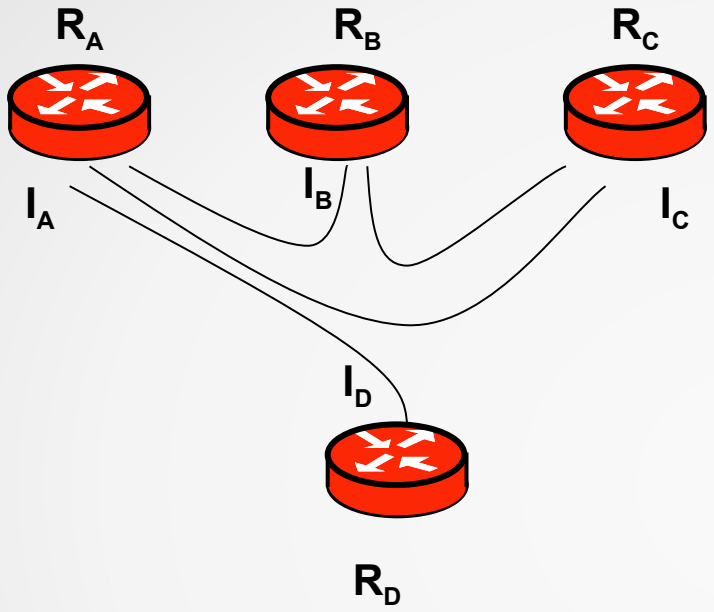



Desde

| Hacia | A | B | C | D | N1 |
|---|---|---|---|---|---|
| A | | | | | 0 |
| B | | | | | 0 |
| C | | | | | 0 |
| D | | | | | 0 |
| N1 | 1 | 3 | 1 | 5 | |

# Redes Punto-Multipunto




Desde

Hacia

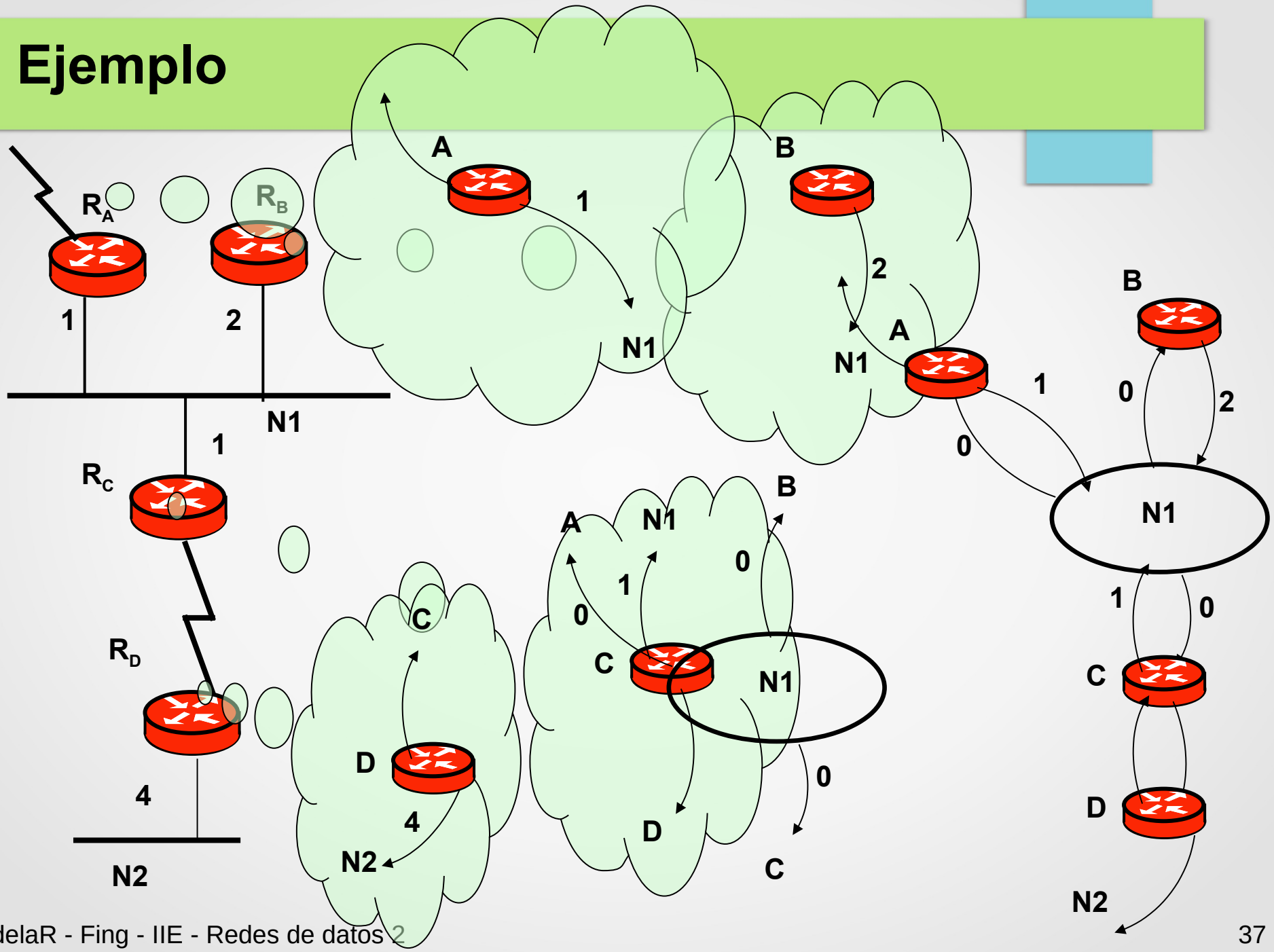
| | $R_A$ | $R_B$ | $R_C$ | $R_D$ |
|---|---|---|---|---|
| $R_A$ | | X | X | X |
| $R_B$ | X | | X | |
| $R_C$ | X | X | | |
| $R_D$ | X | | | |
| $I_A$ | X | | | |
| $I_B$ | | X | | |
| $I_C$ | | | X | |
| $I_D$ | | | | X |

**Se utilizan las redes punto-multipunto en general cuando no se tiene conectividad "todos con todos"**

**Ejemplo: múltiples PVC Frame-relay o ATM (no malla completa)**

**Se trata como múltiples links punto a punto**

# Ejemplo


A
B
1
2
N1
$R_A$
$R_B$
1
2
N1
1
$R_C$
$R_D$
4
N2
C
D
4
N2
A
N1
B
0
1
0
C
N1
D
0
C
A
N1
B
0
1
2
0
1
N1
1
0
C
D
N2

# Cálculo de la mejor ruta

- De los LSA recibidos se obtiene la lista de todas las adyacencias.
  - Se construye el grafo  en base a ellas
- Teniendo el grafo de la red, cada nodo aplica el algoritmo de Dijkstra
- Se obtiene un árbol, con raíz en el nodo, de los caminos de menor peso a todos los nodos de la red
- Del árbol se extraen las mejores rutas a cada destino
- En la tabla de enrutamiento se pone solamente el próximo salto

# Métrica en OSPF

- Valor sin dimensión, entero
- Se calcula utilizando 24 bits
- Infinito: 0xffffff
- El costo de cada interfaz se representa con 16 bits
- Asignado administrativamente
- Por defecto, los equipos asignan un costo
  - Ej. Cisco IOS: 100 Mbps/bw
  - Ej. Cisco Nexus: 40.000 Mbps/bw

  bw: ancho de banda nominal de la interfaz
- Hoy en día es común tener que cambiar el valor de referencia

# Enrutamiento Jerárquico

- Objetivos:
  - Simplificar la tabla de rutas
    - Mediante la agregación en la frontera entre “zonas”
  - Simplificar la topología que debe conocer cada enrutador
- Dificulta algunas aplicaciones (ingeniería de tráfico)

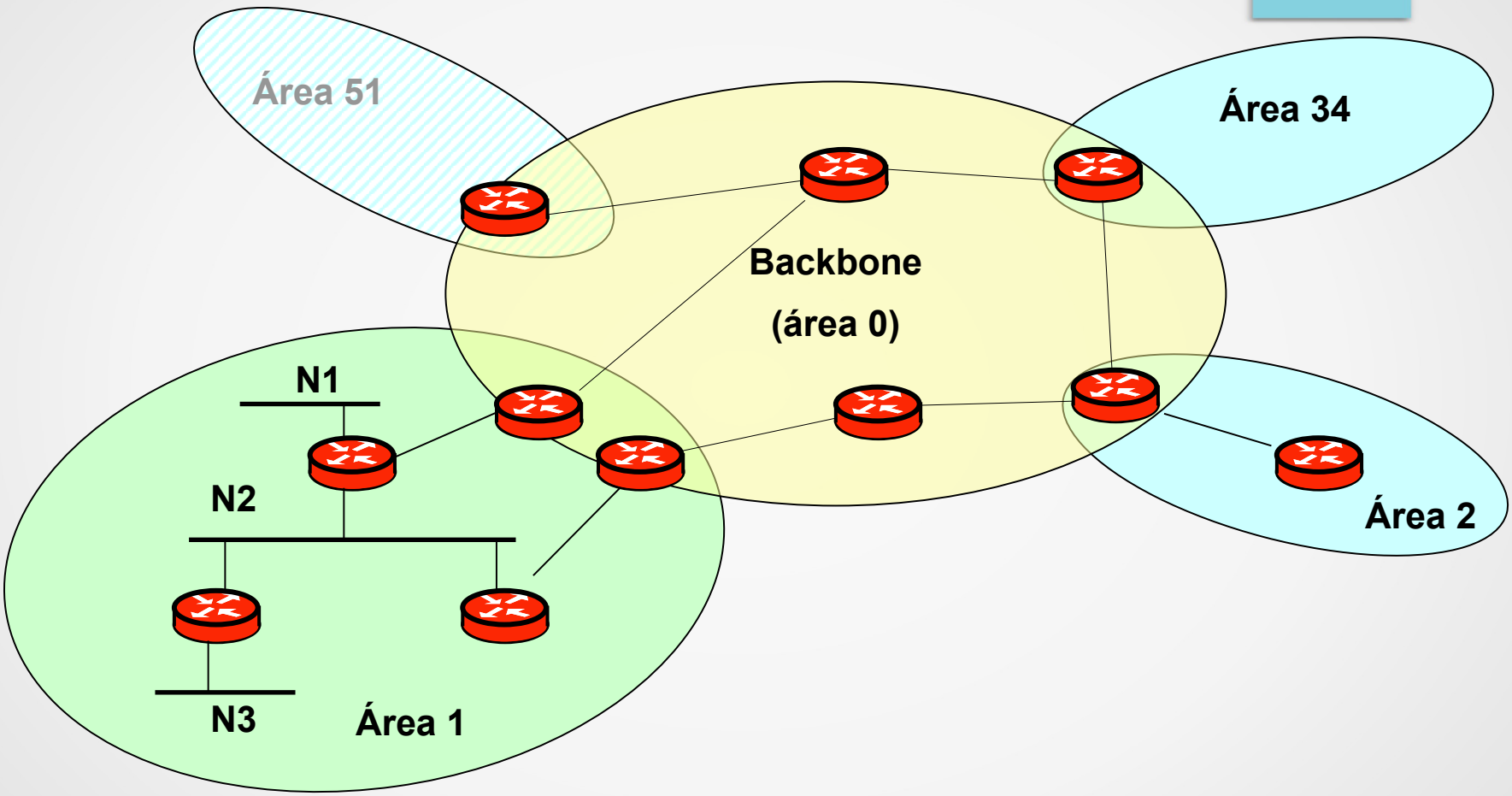
# Enrutamiento Jerárquico

- En OSPF, 2 niveles
- Basado en áreas
- Identificación de Área: número de 32 bits
- Área 0: backbone
- El tráfico entre áreas debe pasar por el backbone
- El backbone debe ser conexo (debe haber una única “area 0”)

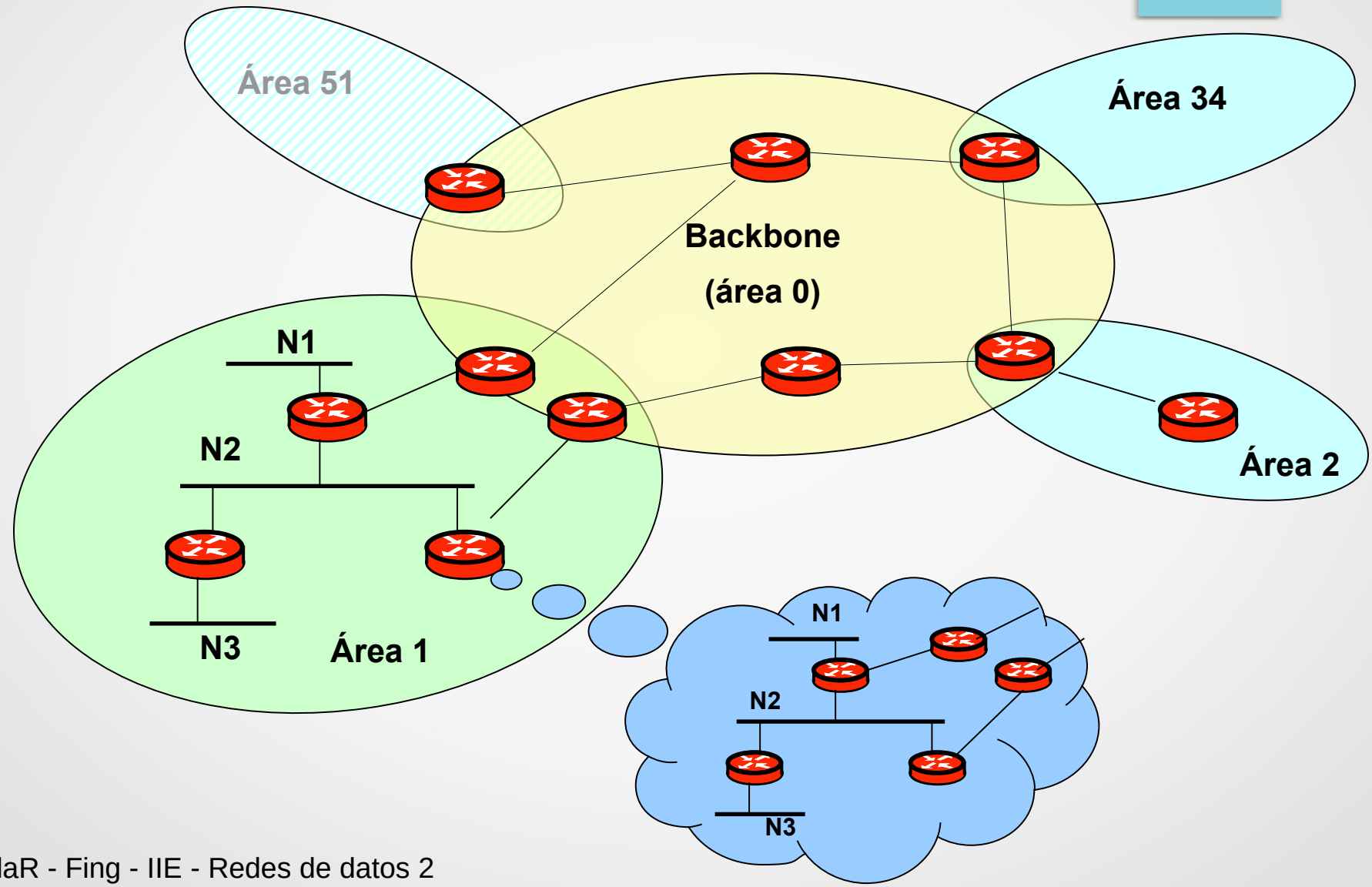
# Ruteo jerárquico

- Enrutadores en un área no conocen la topología externa al área
- La topología del área no se conoce fuera de ella
- Los enrutadores tienen una base de datos de estado de enlace por cada área a la que estén conectados
- Entre áreas solo se propaga las redes alcanzables y la métrica para alcanzarlas

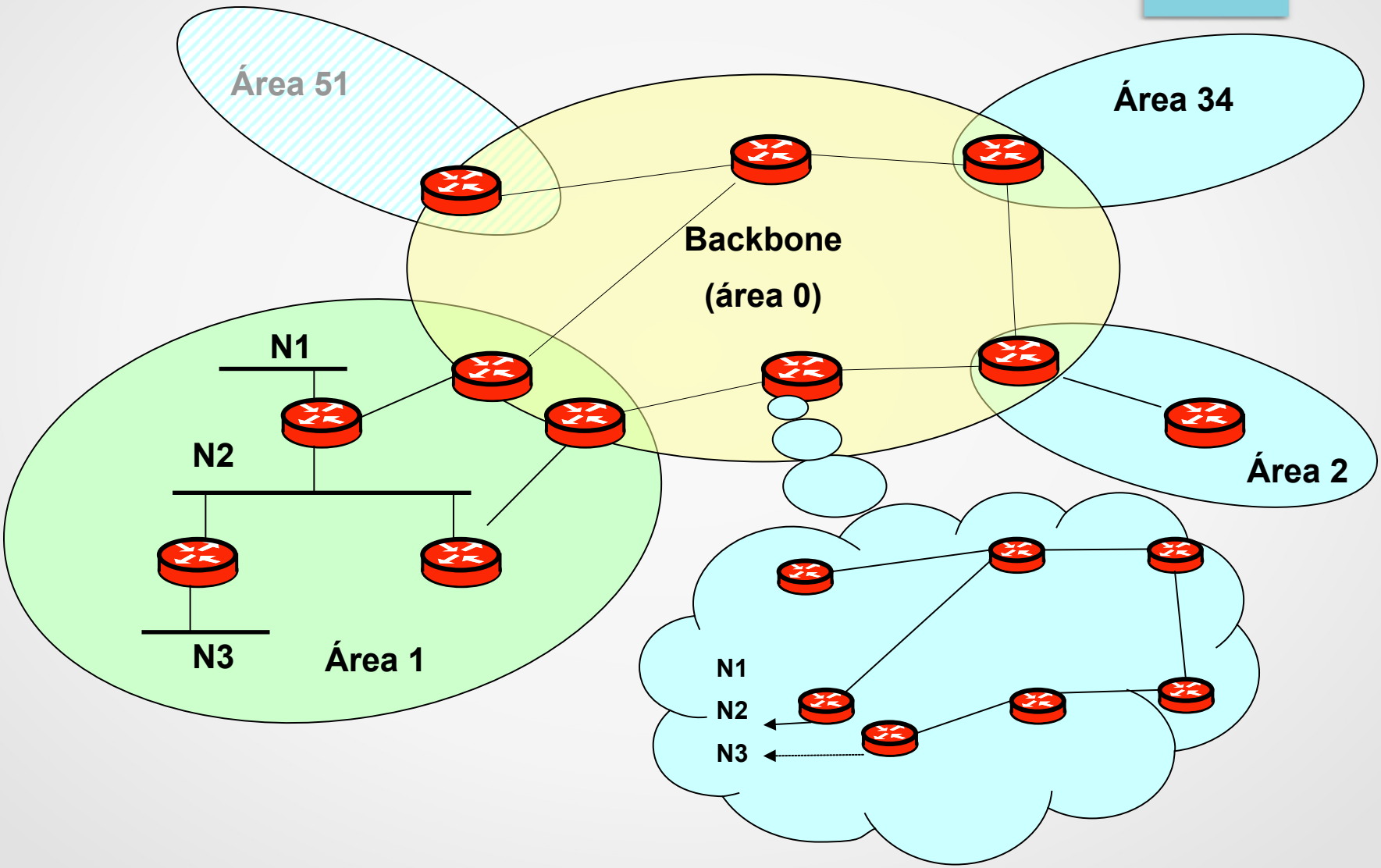
# Visión topológica de las distintas áreas



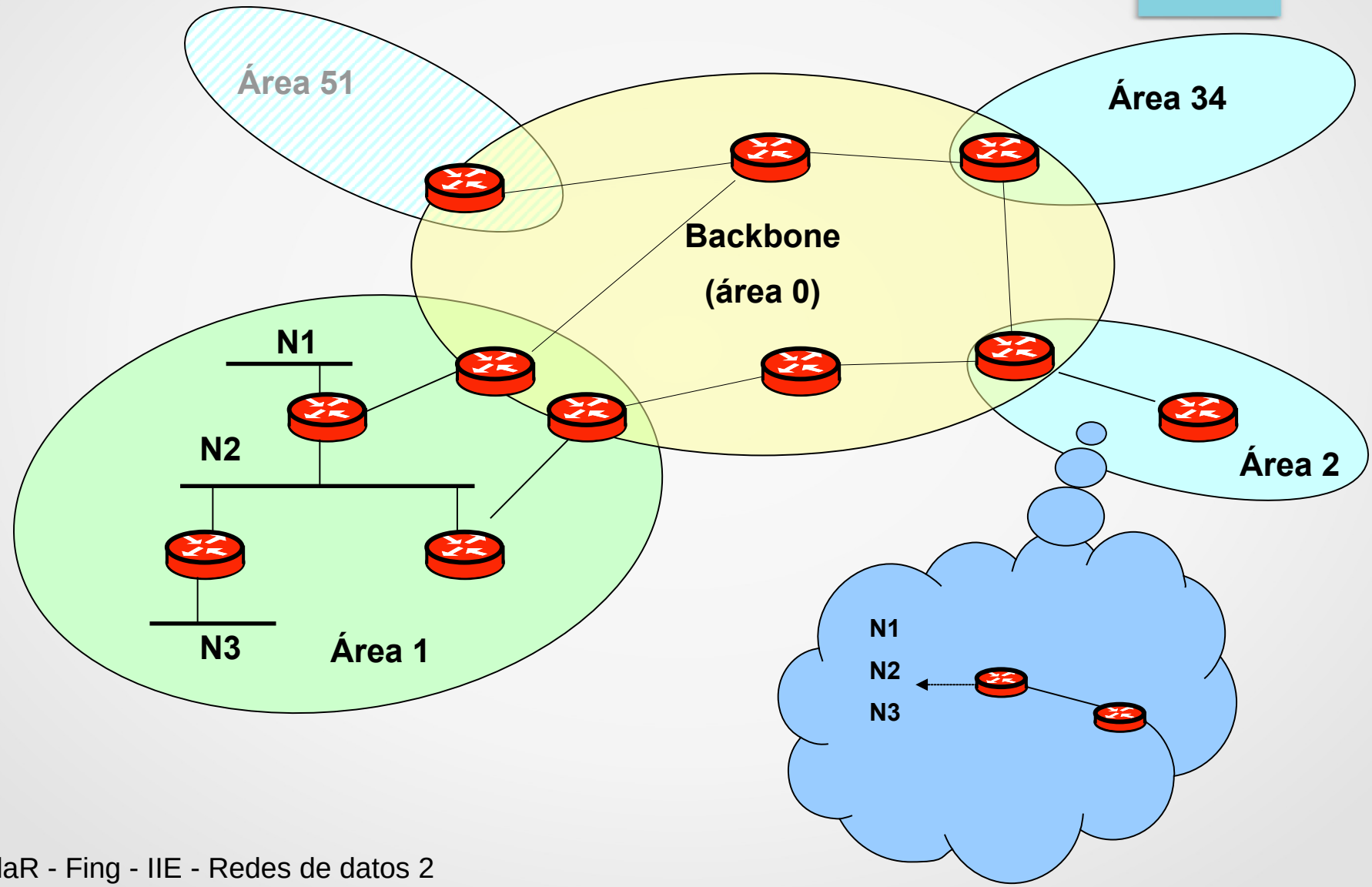
# Visión topológica de las distintas áreas



# Visión topológica de las distintas áreas



# Visión topológica de las distintas áreas



# Clasificación de enrutadores

- Enrutadores internos
  - Pertenecen a una sola área
- Enrutadores de borde de área
  - Pertenecen a más de un área
- Enrutadores de backbone
  - Todos los del área 0 (incluye los de borde de área)
- Enrutadores de borde de AS (sistema autónomo)
  - Inyectan en OSPF información proveniente de otros protocolos, por ejemplo BGP

# Caminos inter áreas

- 3 tramos:
  - Del origen al enrutador de borde de área
  - En el backbone entre enrutadores de borde de área
  - Interno al área de destino

# ¿Por qué la comunicación entre áreas debe pasar por el backbone?

- Entre áreas no se intercambia información de topología
- Si permitiéramos comunicación entre áreas arbitrarias, podríamos tener los mismos problemas de conteo a infinito que con vector distancia
  - Al pasar por el área 0, forzamos una estructura de árbol sin loops

# Continuidad del backbone

- Si el backbone es discontiguo: áreas no podrán comunicarse
  - Las áreas distintas de “0” no hacen tránsito
- Solución: Links virtuales
  - Enlace virtual punto a punto a través de un área no backbone
- Área de tránsito: si un link virtual pasa por ella

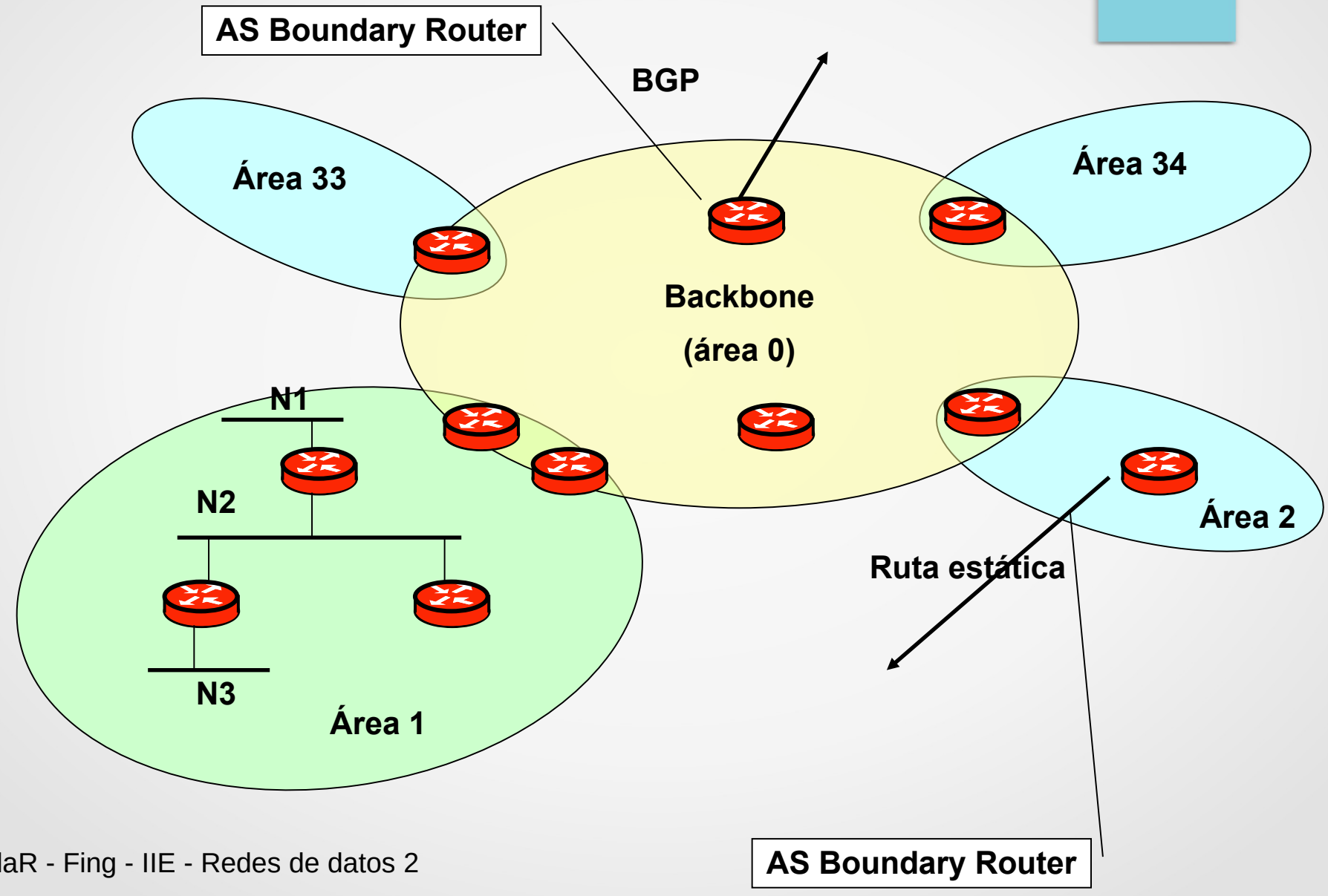
# Agregación

- Puede configurarse (en el enrutador de borde de área) los rangos de IPs del área
- Se puede configurar que fuera del área, solo se propague la ruta correspondiente al rango configurado, y no las rutas específicas

# Información externa

- Proveniente de otros protocolos (o estática)
- Puede incluir ruta por defecto
- 2 tipos según la métrica:
  - tipo 1: métrica comparable con el costo de OSPF
  - tipo 2: métrica estrictamente mayor que cualquier métrica de OSPF
- Se inunda a todo el dominio OSPF

# Información externa



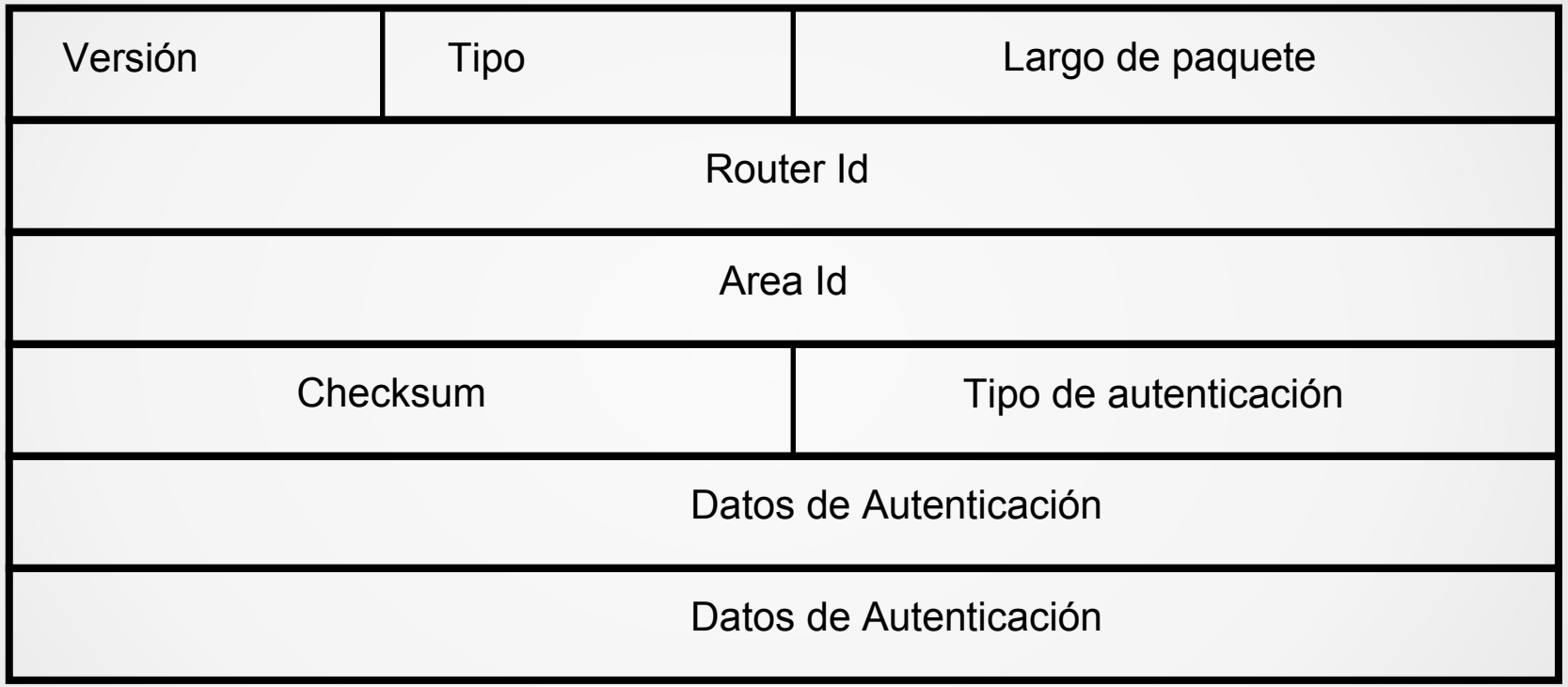
# Paquetes OSPF v2

- OSPF funciona directamente sobre IP
  - Protocolo 89
- En aquellas redes que lo permiten, se utiliza multicast para el envío de mensajes
  - Redes con capacidad de broadcast (ej. Ethernet)
  - Redes punto a punto
  - Se evita configurar los vecinos
- En aquellas redes sin capacidad de multicast, deben configurarse los vecinos
- 2 direcciones de multicast reservadas para OSPF
  - 224.0.0.5 – AllSPFRouters
  - 224.0.0.6 – AllDRRouters

# Tipos de mensaje OSPF

- 5 tipos de paquete, con un encabezado común
  - Hello
  - Database Description
  - Link State Request
  - Link State Update
  - Link State Ack

# Encabezado OSPF
## Común a todos los paquetes



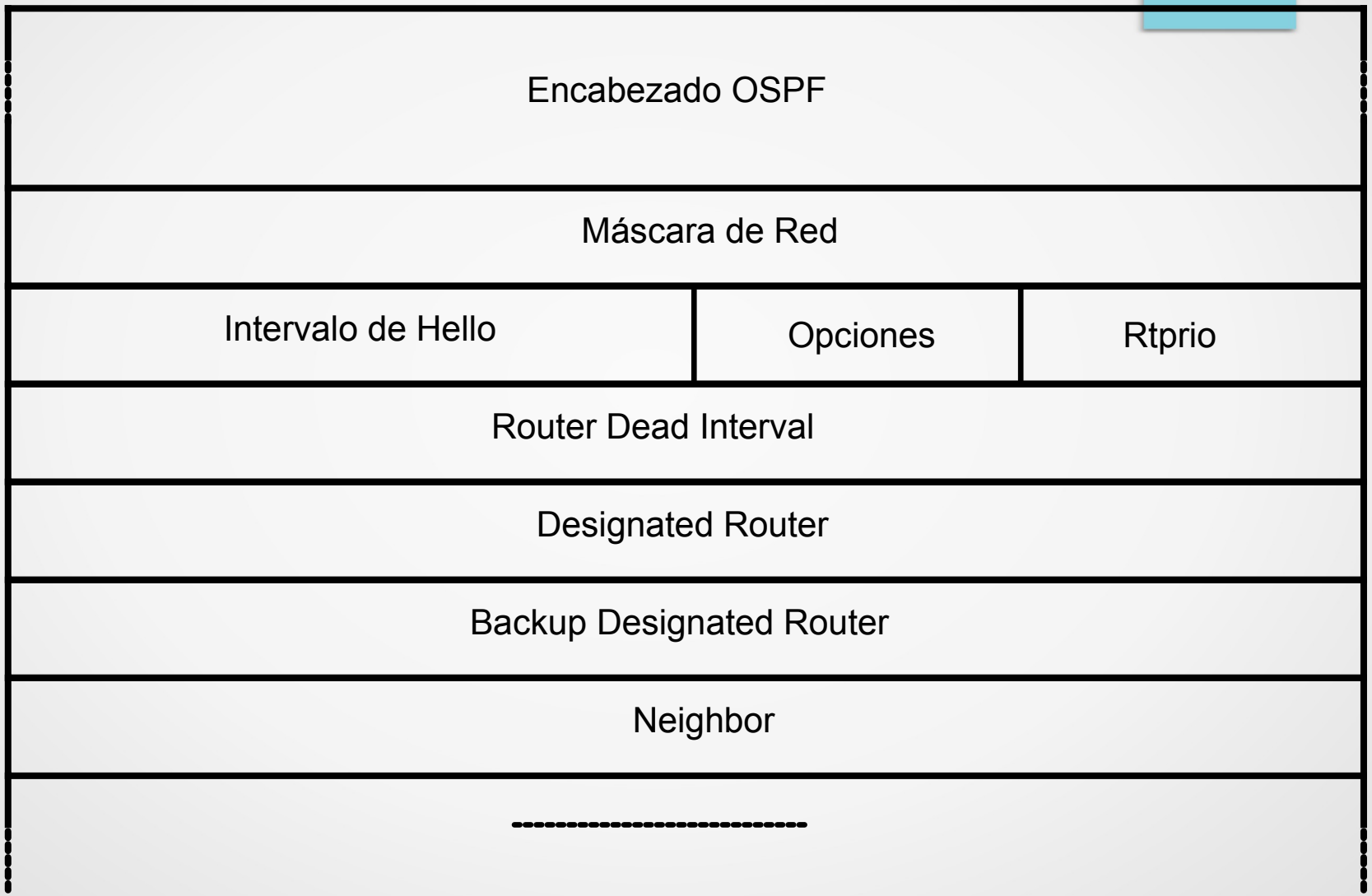
# Autenticación OSPF

- 3 tipos definidos:
  - Null authentication (No autentica!!!)
  - Simple password (password en claro!!!)
    - Se compara la password en el paquete con la configurada
  - Cryptographic authentication
    - Se envía un “message digest” (ej. MD5, SHA1) de la concatenación de mensaje + un secreto compartido
    - En el campo de autenticación se describe la clave, largo del digest, y se agrega un número de secuencia (para evitar replay)

# Protocolo HELLO

- Envía paquetes de descubrimiento (HELLO) periódicamente (por defecto HelloInterval=10s)
- Descubrimiento de vecinos
- Anuncio de parámetros
- Establecimiento de comunicación bidireccional
- Generación de adyacencias
- En redes Broadcast y NBMA, elige DR
- Keepalive

# Paquete de Hello



# Campo de Opciones

- DC bit: soporte de OSPF en circuitos a demanda
- EA bit: Soporte de External Attributes (type 8) LSAs
- N bit: Soporte de NSSA LSAs
- P bit: ABR debería traducir LSAs tipo 7 en tipo 5
- MC: MOSPF (multicast)
- E: Soporte para AS External LSAs. Apagado en areas STUB
- T: Soporte de TOS

# HELLO (cont.)

- En redes Punto a punto y broadcast:
  - Enviado a 224.0.0.5 (AllSPFRouters)
- En redes sin capacidad de Multicast (NBMA): unicast (debe conocer los vecinos)
- Formación de Vecindades:
  - Solamente si los parámetros en el HELLO son iguales (máscara, RouterDeadInterval, Intervalo de Hello, área, Autenticación, Opciones)
    - (si no se forma una adyacencia, revisar parámetros!!!)

# Hello (cont.)

- Al recibir el mensaje Hello, si aparezco como vecino, sé que la comunicación es bidireccional
- Adyacencia: comunicación punto a punto entre ciertos vecinos
- Se rompe si no recibo Hello en RouterDeadInterval (por defecto 4 * HelloInterval)

# DR: Designated Router

- Solo en redes Broadcast y NBMA
- Se elige un router para propagar el LSA de la red de la LAN y minimizar anuncios
- Es adyacente a los demás routers de la LAN
- También se elige un backup D.R.
- DR y Backup DR mandan paquetes a 224.0.0.5 (AllSPFRouters)
- Los demás a 224.0.0.6 (AllDrRouters)

# Elección del D.R.

- Al ingresar un nuevo enrutador:
  - Si los paquetes HELLO recibidos contienen un DR y BDR, aceptarlos
  - Si no hay DR elegido,  router con mayor prioridad se convierte en DR, y el de segunda mayor prioridad BDR
  - Si no hay BDR elegido, router con mayor prioridad se convierte en BDR
  - Empate se define con el mayor Router ID
  - Si se pierde el DR, el BDR es promovido a DR
    - Se elige BDR nuevamente
- Estados posibles de una interfaz Broadcast: DR, BDR, DROther

# Deshabilitando elección del DR

- Hoy en día, es común tener enlaces ethernet “punto a punto”
  - Dos enrutadores conectados directamente con una capa de enlace Ethernet
- Elección del DR y generación de LSA de la red superfluos
- En estos casos (si el equipo lo permite) podemos configurar las interfaces como “punto a punto”
  - CUIDADO, deben estar configurados igual en ambos extremos

# Formación de adyacencias

- Líneas punto a punto: los vecinos siempre se convierten en adyacentes
- Redes Broadcast/NBMA: todos los routers se convierten en adyacentes del D.R. y el Backup D.R.
- La información de estado de enlace se intercambia entre nodos adyacentes

# Sincronización inicial de la base de datos

- Solo al inicializarse una adyacencia
- Relación maestro/esclavo (router con mayor router-id es maestro)
- Vecinos intercambian resúmenes de los LSA que disponen (Database Description Packets)
  - Age, Options, Type (of LSA)
  - Link State Id.
  - Advertising Router, Sequence Id.

# Sincronización (cont.)

- Determinan qué LSAs no tienen o tienen desactualizados
- Solicitan los LSA faltantes
  - Paquetes de LSA Request
- Son enviados en LSA Updates
- Deben ser reconocidos (LSA Ack)

- Una vez completado, la adyacencia está funcional

# Comparación de LSAs

- Entre distintos elementos de la red, comparo la identificación y el tipo
- Importante poder determinar si 2 LSA recibidos correspondientes al mismo elemento de la red corresponden a la misma información, o si uno de ellos tiene información más reciente
- Objetivo primordial: que todos los enrutadores tengan la misma visión de la topología de la red

# Comparación de LSAs: datos disponibles en cada LSA

- Número de secuencia: 32 bits con signo
  - Se incrementa con cada actualización
- Edad: en segundos, inicializada en 0
  - Entero sin signo (16 bits)
  - Se incrementa al reenviar y con paso del tiempo
- Edad máxima: MaxAge = 3600
- El LSA se descarta si pasa de MaxAge
  - Antes de descartar se inunda nuevamente

# Algoritmo de comparación de LSAs

- Número de secuencia mayor es más nuevo
- Mayor checksum es “más nuevo”
  - Desempate (*)
- Si tiene AGE en MaxAge, es más nuevo (**)
- Si las edades difieren en más de MaxAgeDiff, la menor es más nueva
- De lo contrario son iguales

# Comentarios

- (*)Si recibimos 2 LSA que deberían ser iguales, con distinto checksum, uno de ellos “está mal”. Lo importante es que todos seleccionen el mismo
  - Quien lo origina se dará cuenta y lo generará nuevamente
- (**)Considerar más nuevo un anuncio expirado, tiene como consecuencia que rápidamente todos los enrutadores descarten ese anuncio. Se utiliza como mecanismo para “borrar” un LSA

# Inundación confiable de LSAs

- Se hace enlace a enlace
- Mediante paquetes de “Link State Update”
- Cada paquete de actualización puede contener varios LSAs
- Cada LSA debe ser reconocido
- Se pueden agrupar varios reconocimientos en un Link State Acknowledge Packet

# Inundación de LSAs (cont)

- Al recibir un LSA más nuevo que la copia disponible (o al generar uno nuevo)
- Se debe inundar por todas las interfaces (excepto por la que se recibió si no somos DR)
  - Se agrega a la lista de retransmisión para cada vecino (que no sepamos que lo tiene)
  - Se inunda en las interfaces con vecinos que no dispongan de la información

# Inundación (cont.)

- En interfaces Broadcast o NBMA:
  - Si somos DR -> se inunda a AllSPFRouters
  - Si no -> se inunda a AllDRRouters
- Se retransmite hasta que veamos un ACK del vecino
  - Si el vecino nos lo retransmite se considera un ACK
- Todo LSA recibido debe ser reconocido

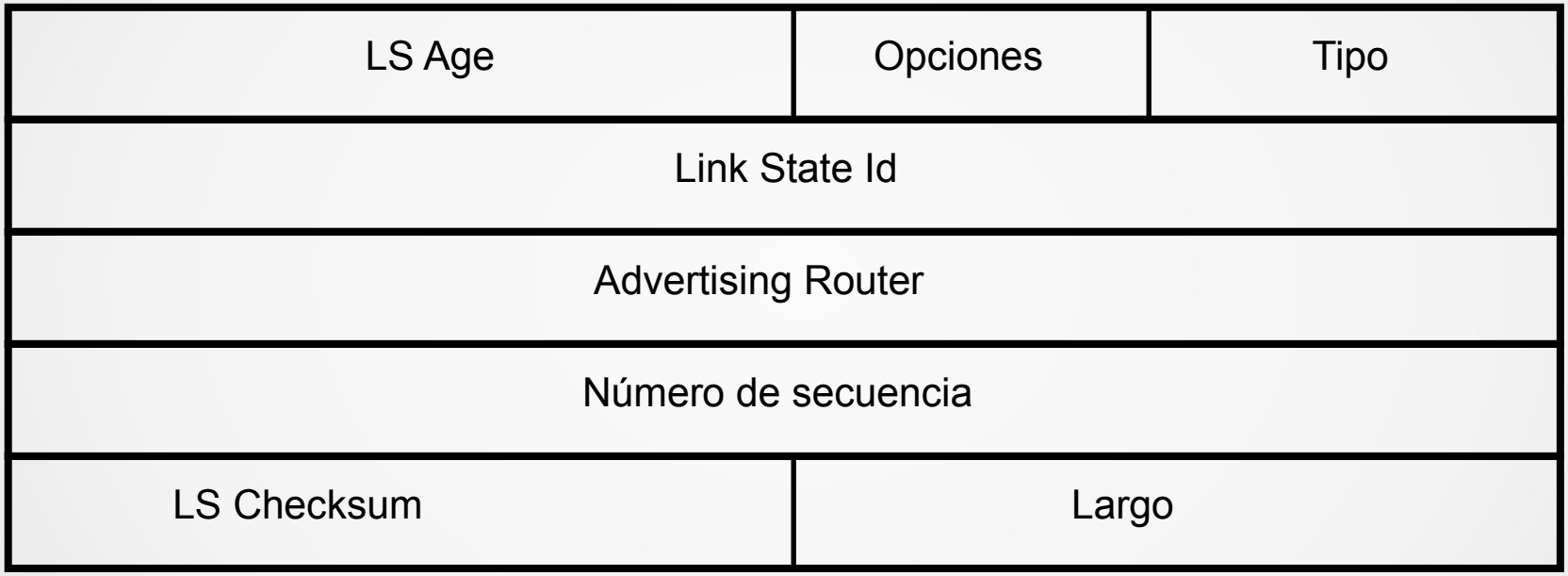
# Tipos de LSA (RFC 2328)

- 1. Router
- 2. Network
- 3. Network Summary
- 4. ASBR Summary
- 5. AS External

# Otros tipos de LSA

- 6. Group Membership
- 7. NSSA External
- 8. External Attributes
- 9. Opaque (link-local scope)
- 10.Opaque (area-local scope)
- 11.Opaque (AS scope)

# Encabezado de un LSA



# Router LSA

- Generado por cada router
- Incluye
  - Identificación del router (Link State Id)
  - Lista de todos los enlaces
    - Identificación de interfaz
    - Tipo de enlace
    - Métrica

# Router LSA


Encabezado LSA
0
V
E
B
0
# links
Link Id
Link Data
Type
#TOS
Métrica
Link Id
Link Data

# Router LSA

- Link State Id (en el encabezado del LSA): Router Id
- V: router tiene links virtuales
- E: router es AS-boundary
- B: área border router

# Router LSA (cont)

- Para cada link:
  - Tipo
    - Punto a punto
    - Transit network
    - Stub network
    - Virtual
  - Link Id (a quién se conecta)
    - Si es a un router o red de tránsito: ID
    - Caso contrario, IP de red

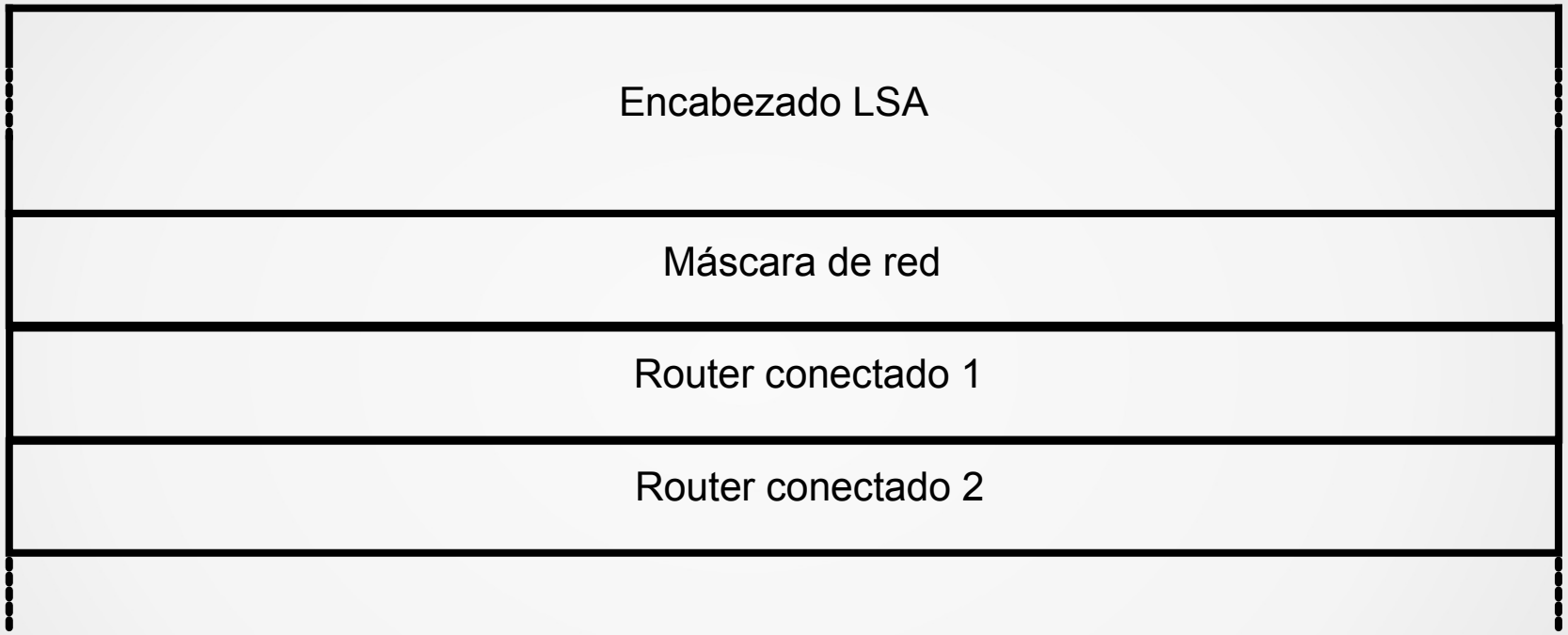
# Router LSA (cont)

- Link data:
  - para redes stub, la máscara
  - para punto a punto no numeradas: identificador de la interfaz
  - para el resto: IP del router en la interfaz
- Métrica

# Network LSA

- Solo para redes multiacceso
- Generado por el DR
- Nodo “virtual” que representa la red multiacceso en el grafo
- Dirección de red y máscara
- Direcciones de los routers conectados

# Network LSA

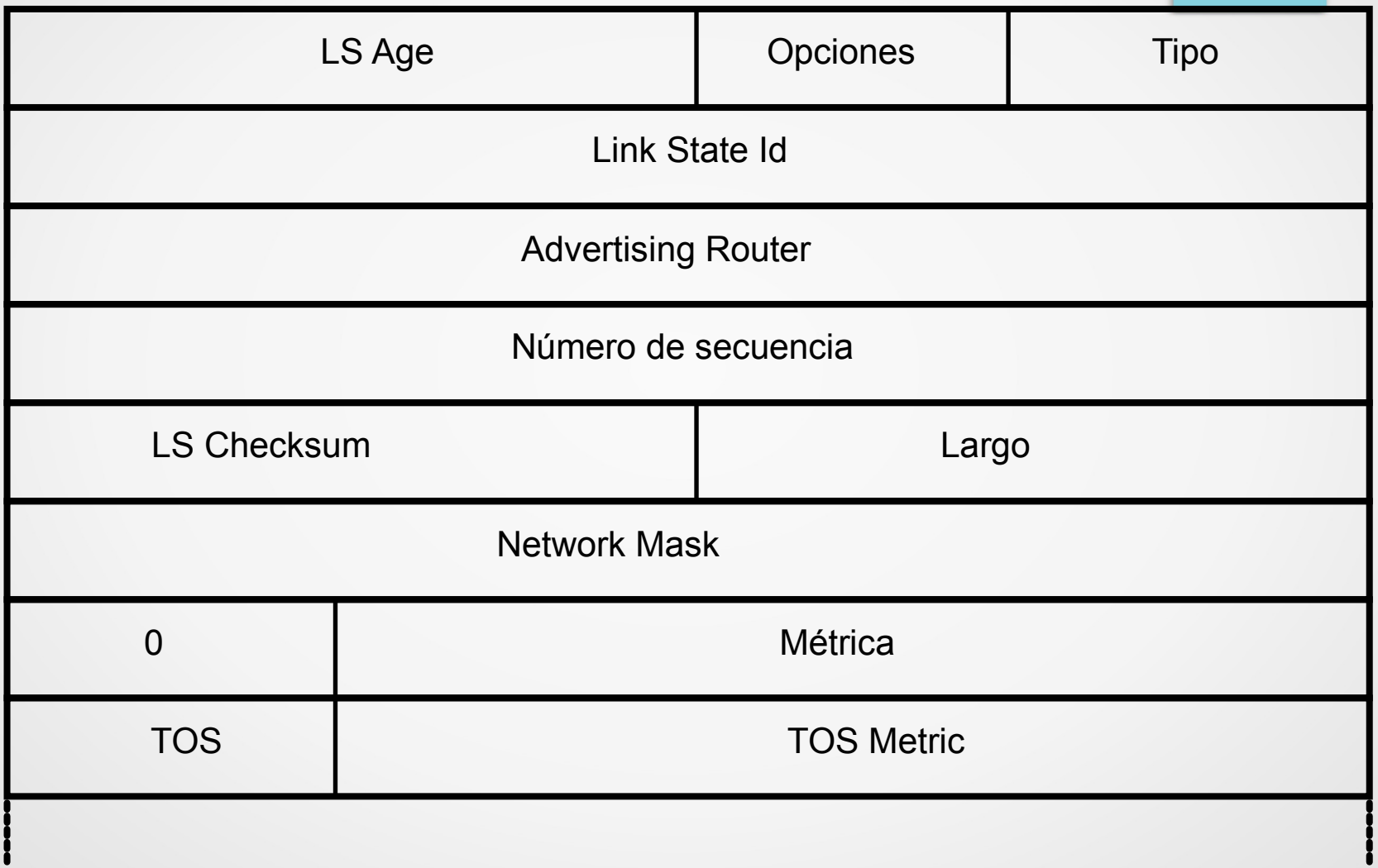



- Router conectado: Router-id de cada uno de los routers adyacentes al DR

# Summary LSAs (Tipos 3 Y 4)

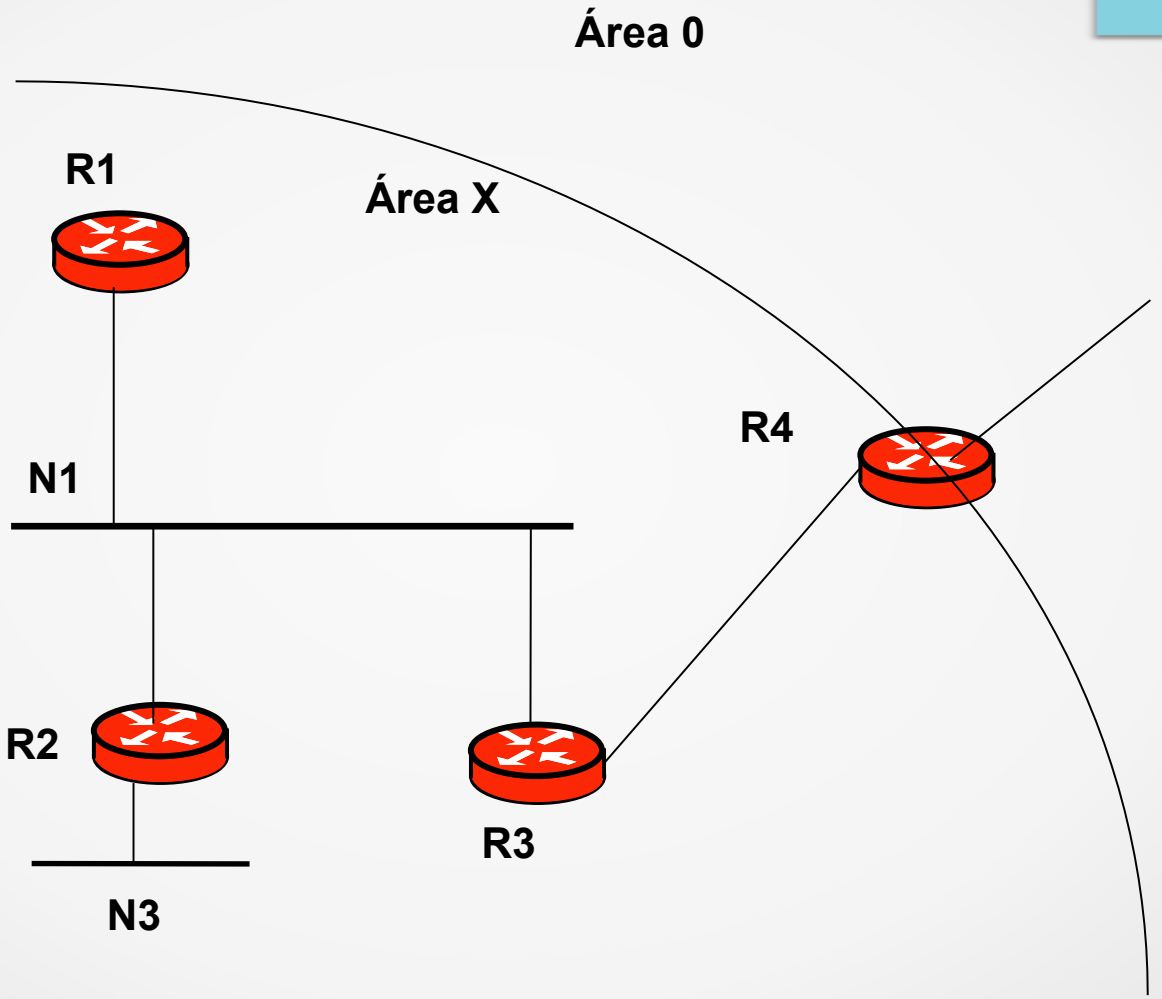
- Tipo 3: descripción de una red hacia otra área
- Link State Id: Dirección de red
- máscara de la red
- métrica
- Tipo 4: descripción de AS boundary router
- Link State Id: Router-id

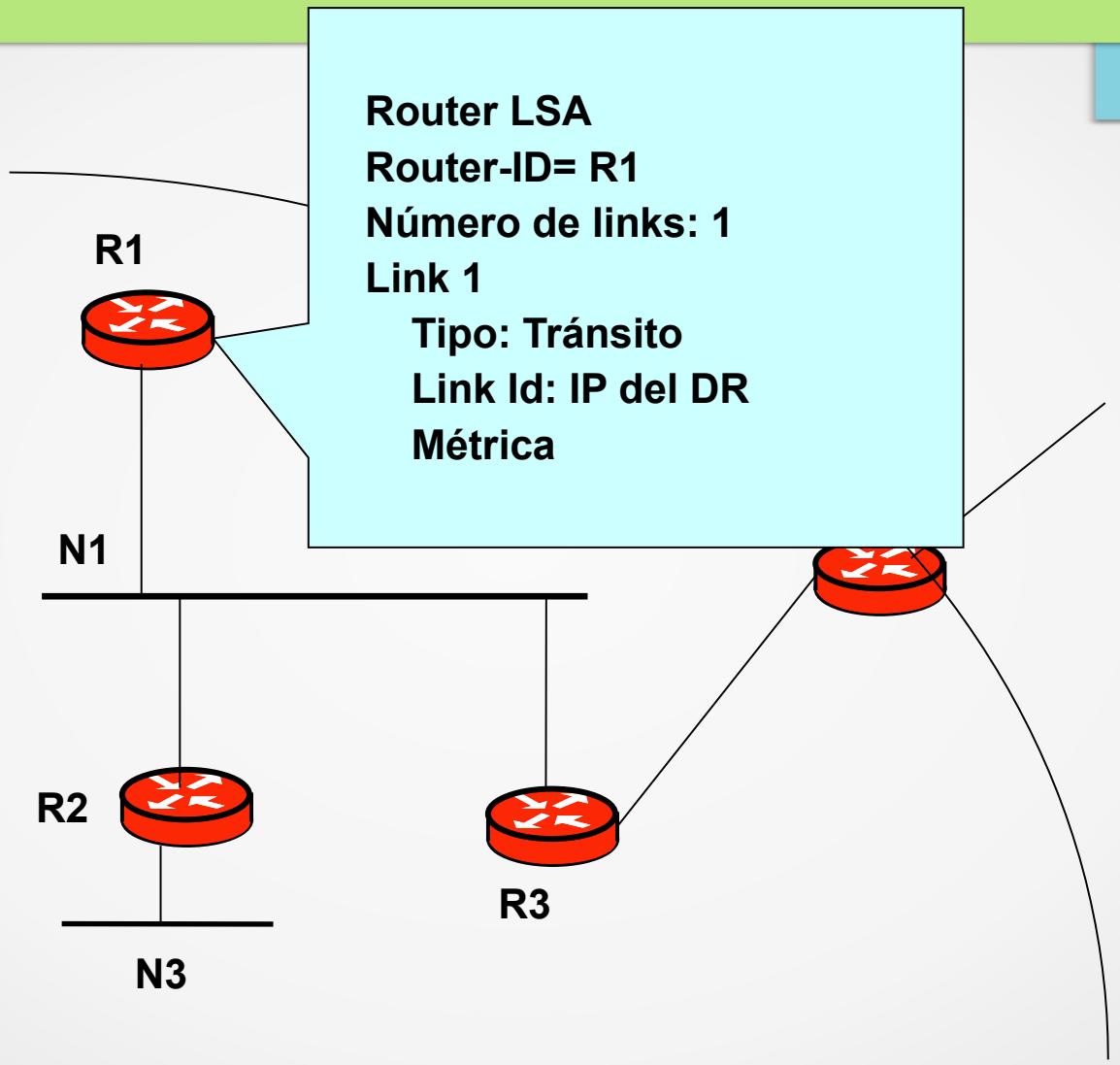
# LSA tipo Summary

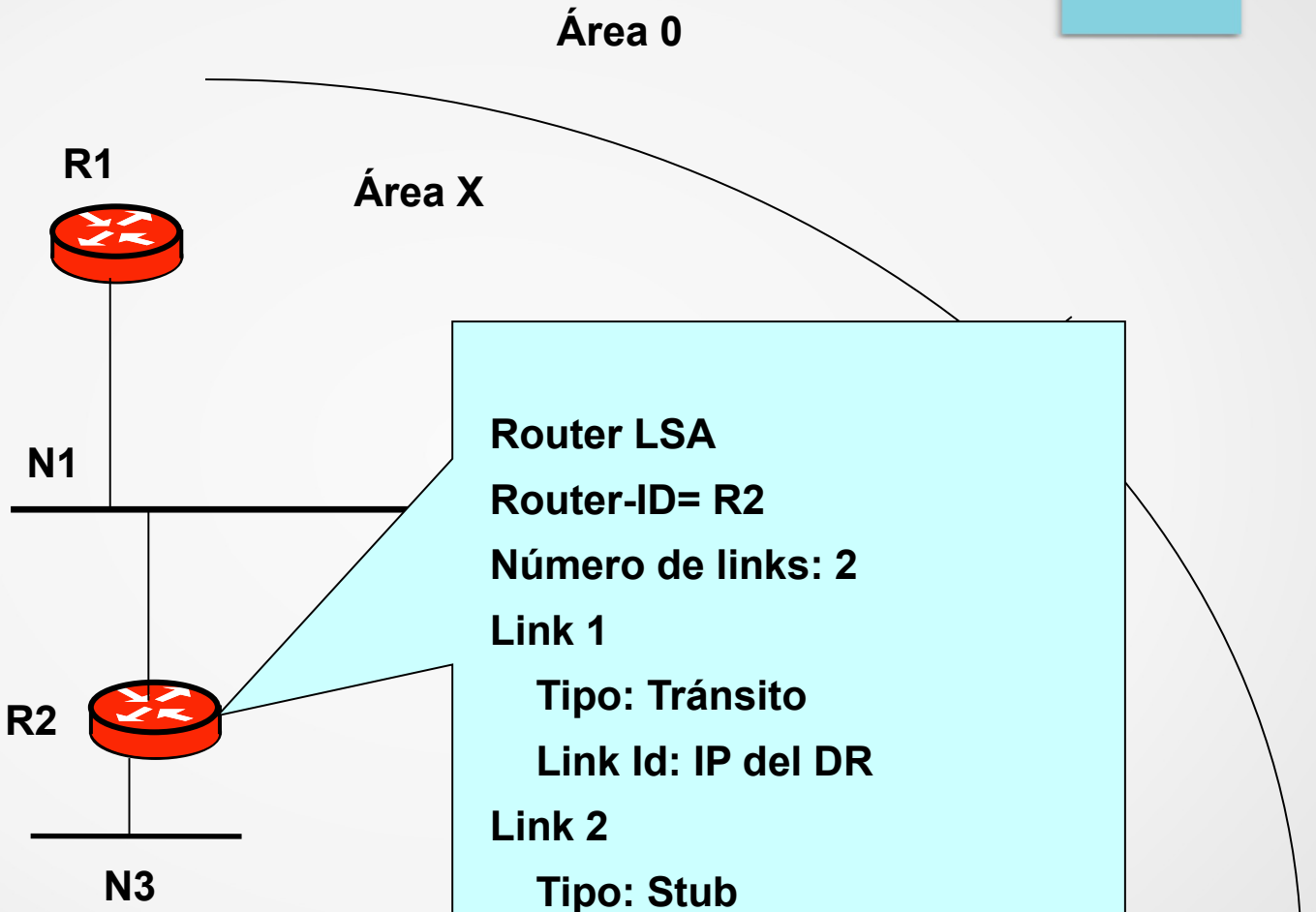


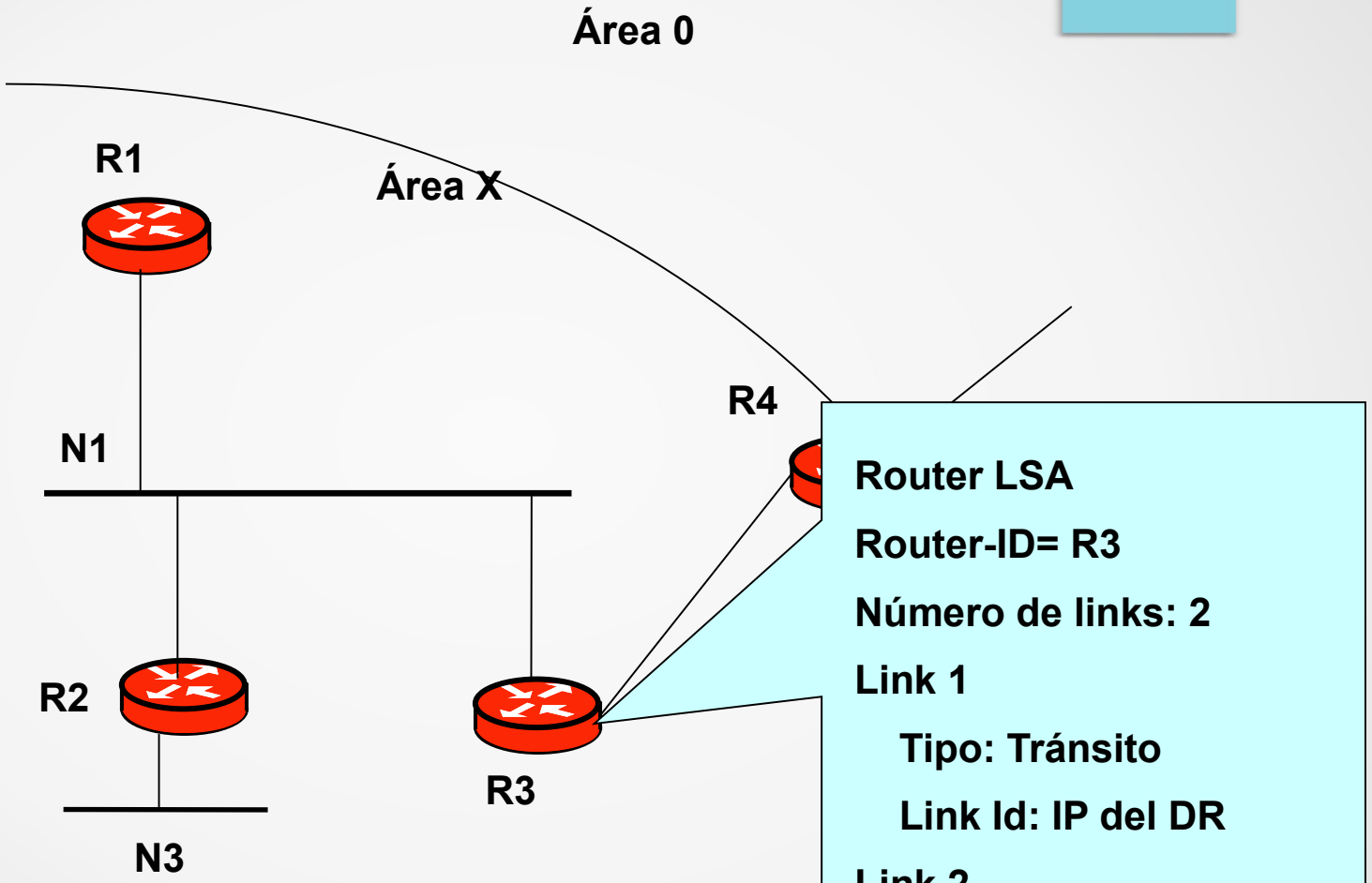
# AS-External-LSA (tipo 5)

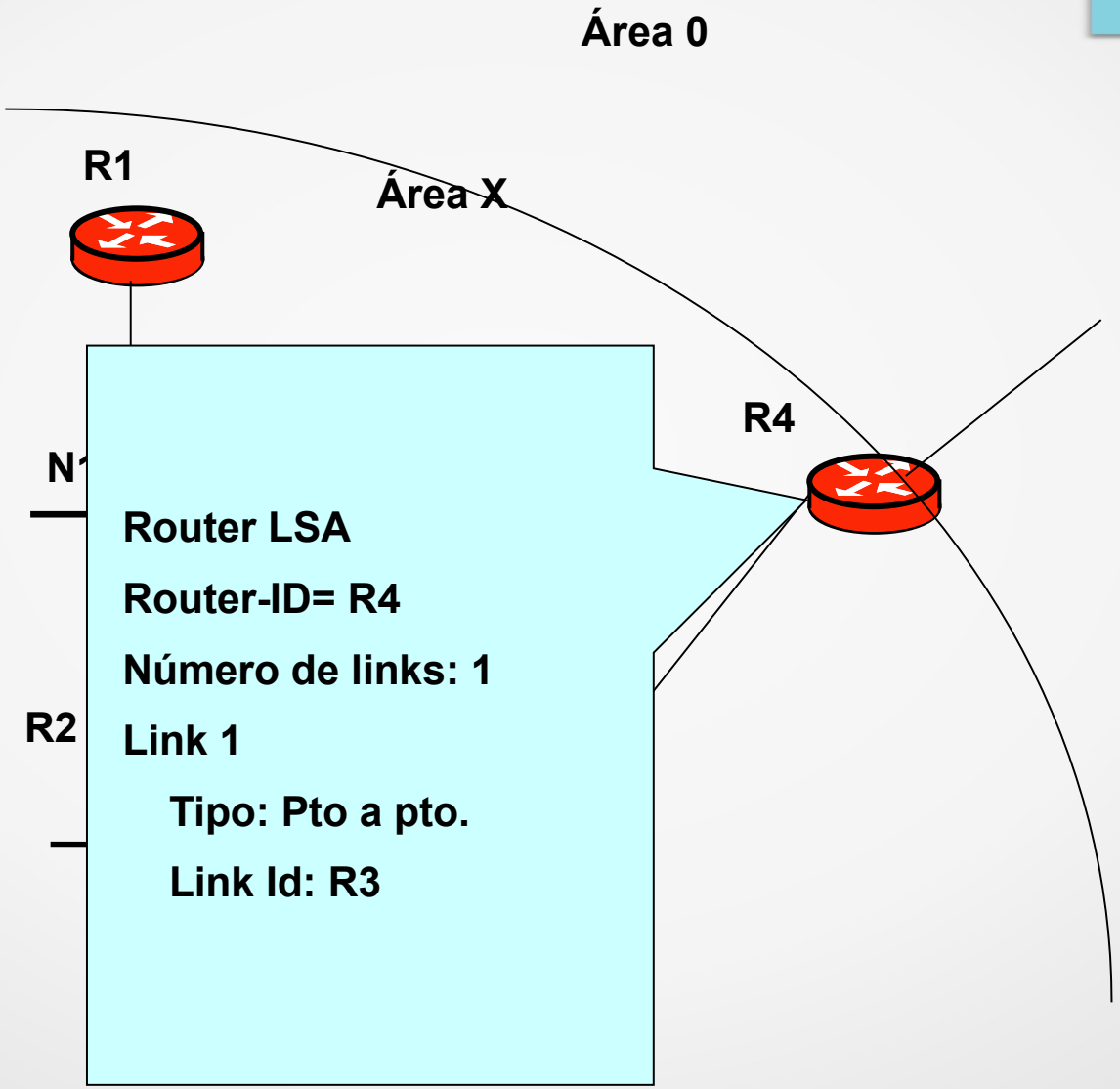
- Describe destinos externos al AS (externos a OSPF)
- Red
- Máscara
- Forwarding address (puede ser distinto del Advertising Router)
- Tipo de métrica externa
  - Tipo 1: comparable con OSPF
  - Tipo 2: estrictamente mayor que métrica OSPF

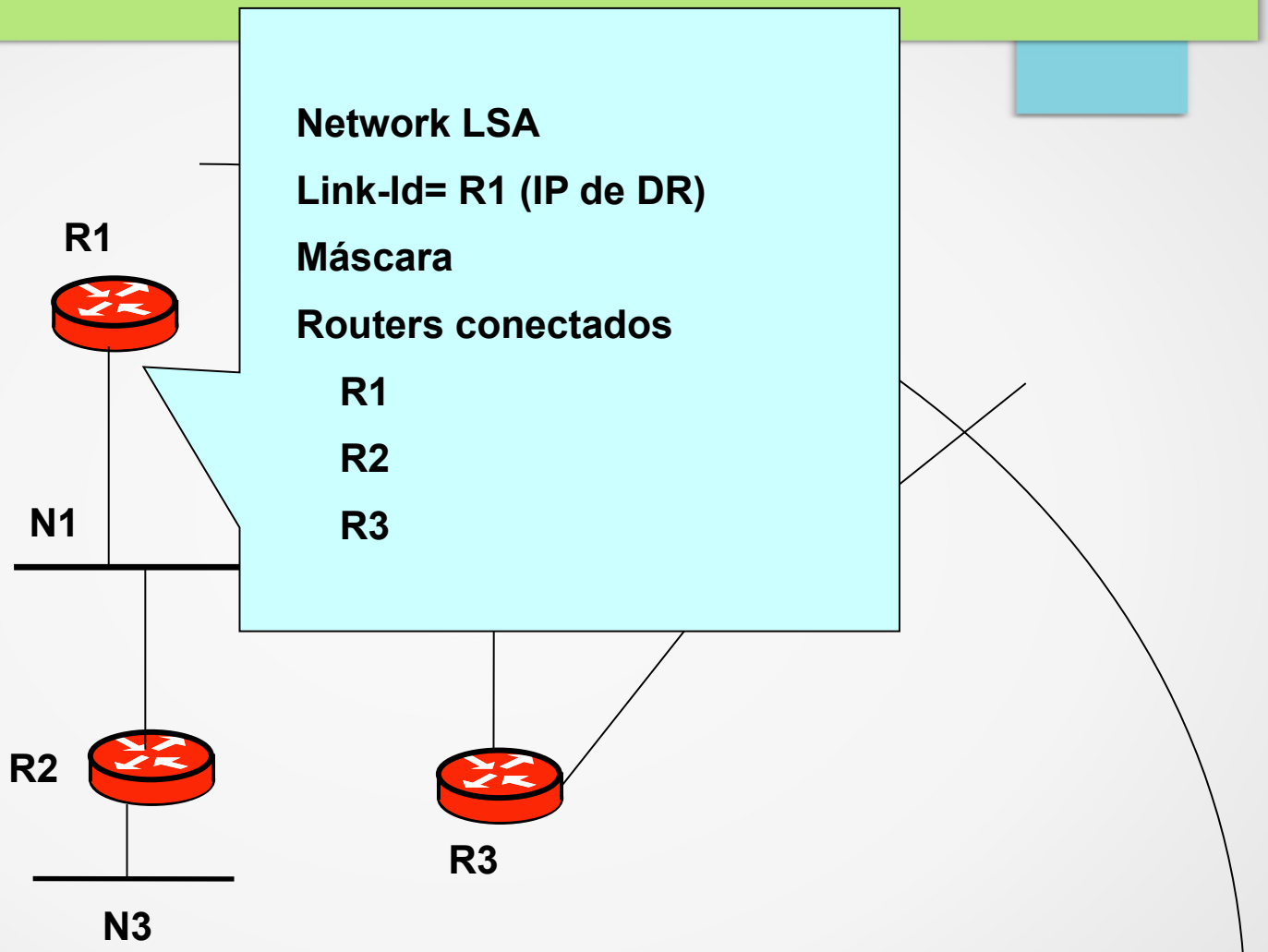

Área 0
R1
Área X
R4
N1
R2
R3
N3

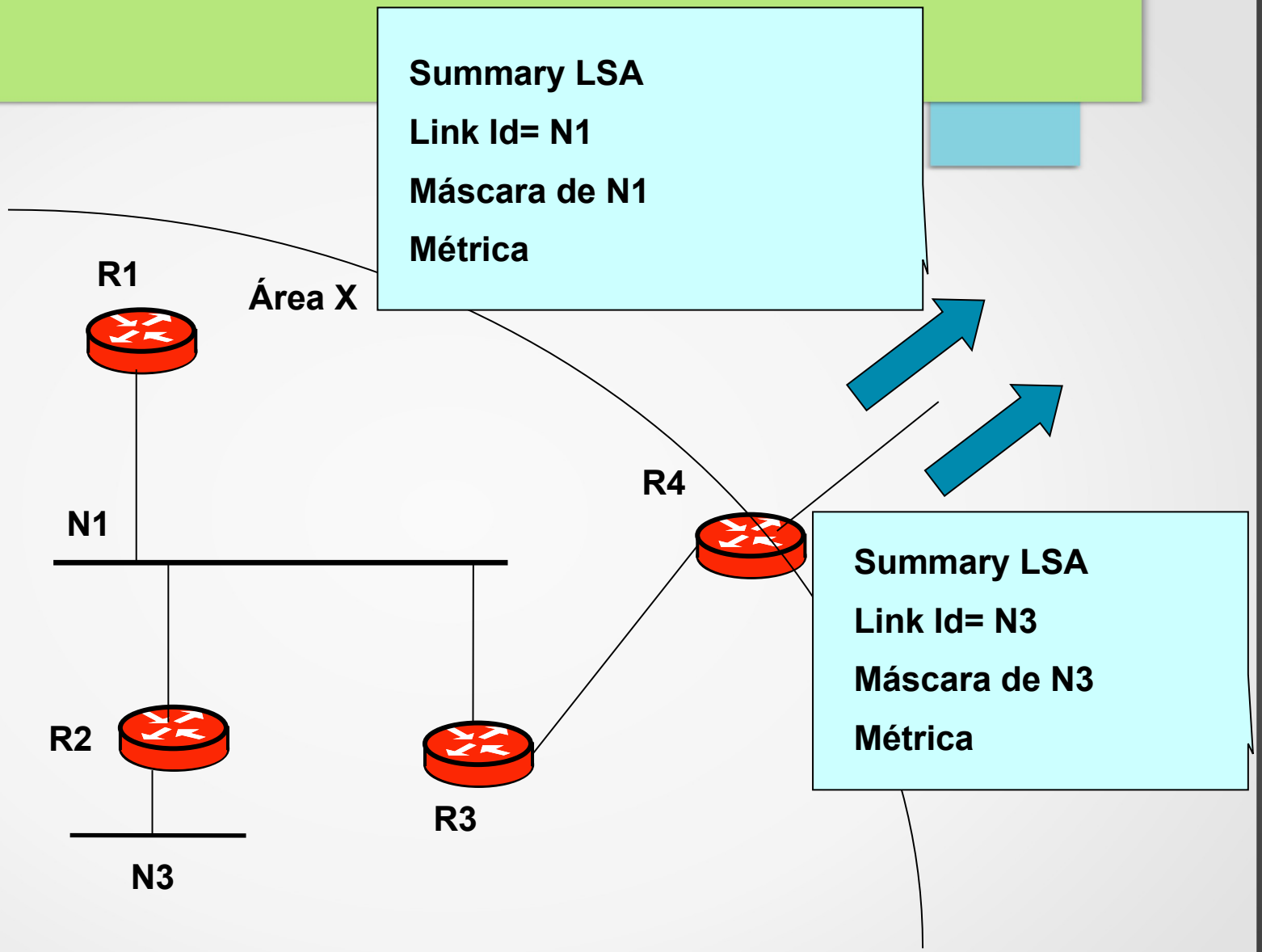

Router LSA
Router-ID= R1
Número de links: 1
Link 1
Tipo: Tránsito
Link Id: IP del DR
Métrica
R1
N1
R2
R3
N3


Área 0
Área X
R1
N1
R2
N3
Router LSA
Router-ID= R2
Número de links: 2
Link 1
Tipo: Tránsito
Link Id: IP del DR
Link 2
Tipo: Stub
Link Id: N3 (IP de red)
Máscara de la red

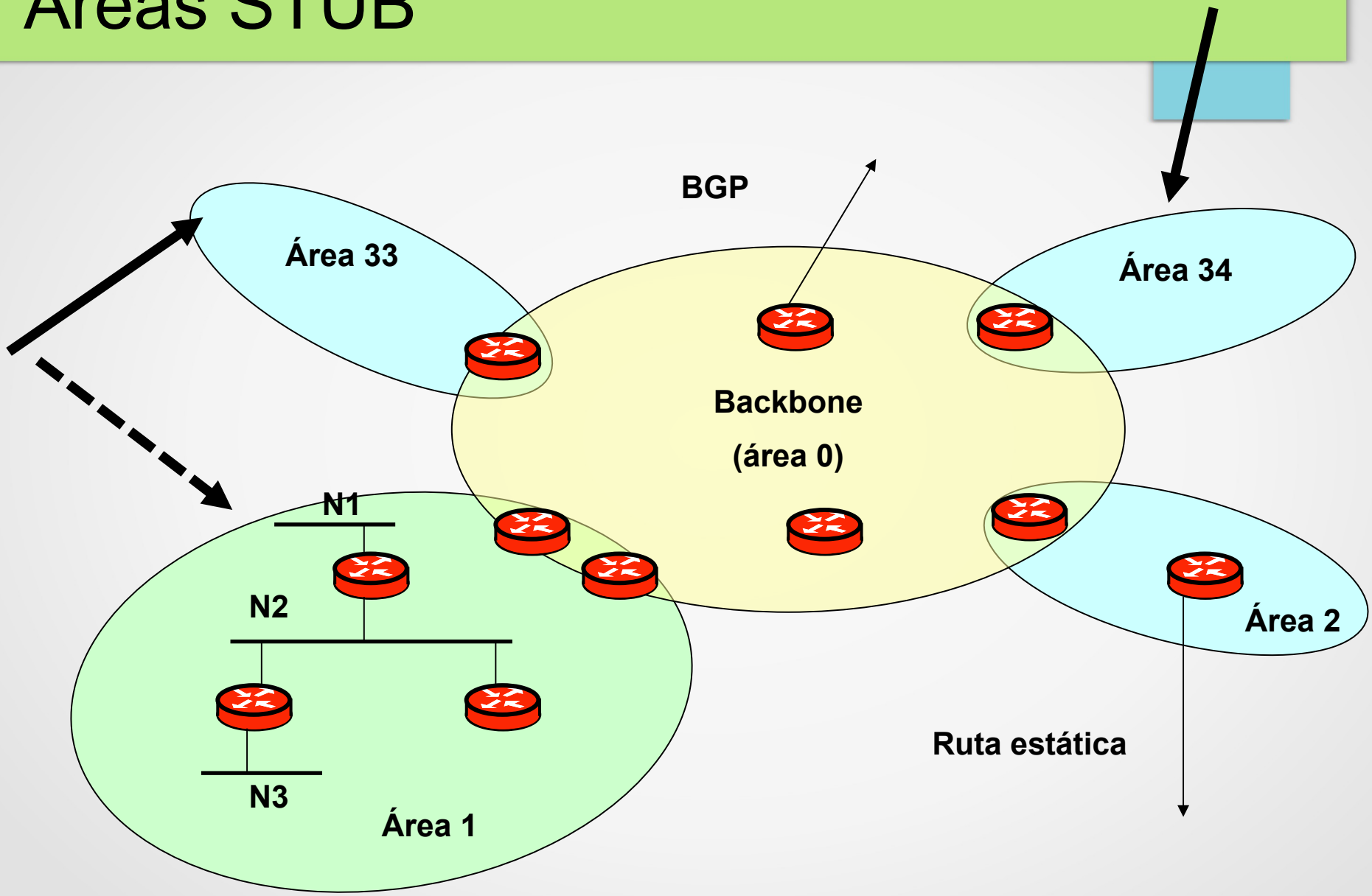

Área 0
Área X
R1
R4
N1
R2
R3
N3
Router LSA
Router-ID= R3
Número de links: 2
Link 1
Tipo: Tránsito
Link Id: IP del DR
Link 2
Tipo: Pto a pto.
Link Id: R4


Área 0
R1
Área X
R4
N1
Router LSA
Router-ID= R4
Número de links: 1
R2
Link 1
Tipo: Pto a pto.
Link Id: R3


Network LSA
Link-Id= R1 (IP de DR)
Máscara
Routers conectados
R1
R2
R3
R1
N1
R2
N3
R3


Summary LSA
Link Id= N1
Máscara de N1
Métrica
R1
Área X
R4
N1
Summary LSA
Link Id= N3
Máscara de N3
Métrica
R2
R3
N3

# Áreas STUB


BGP
Área 33
Área 34
Backbone
(área 0)
N1
N2
N3
Área 1
Área 2
Ruta estática

# Áreas STUB

- “Simplificación” para áreas con “una” conexión al backbone
- Innecesario conocer información externa al AS si tenemos una sola salida del área. Posiblemente tampoco de otras áreas
- Solo se publica una ruta por defecto hacia el área
- No puede haber links virtuales que pasen por un área stub
- No se permiten LSA tipo 5 (external)
- Opción standard: eliminar los LSA de tipo 5 (external)
- Opción de fabricantes: eliminar también LSA Summary

# LSA tipo 7. NSSA External

- NSSA - Not So Stubby Area
- Objetivo: generar información externa en un área STUB
- Se utiliza un nuevo tipo de LSA (tipo 7), ya que el 5 está prohibido en áreas STUB
- Saliendo del área STUB, el enrutador de borde puede ser configurado para descartarlo o convertirlo a tipo 5