

Enrutamiento

Link-state...

Area de Ingeniería Telemática
<http://www.tlm.unavarra.es>

Redes
4º Ingeniería Informática

Hoy...

1. Introducción a las redes
2. Tecnologías para redes de área local
3. Conmutación de circuitos
4. Tecnologías para redes de área extensa y última milla
- 5. Encaminamiento (sesión 3)**
 - Arquitectura de conmutadores de paquetes
 - Control de acceso al medio
 - Transporte extremo a extremo

Enrutamiento Link-State

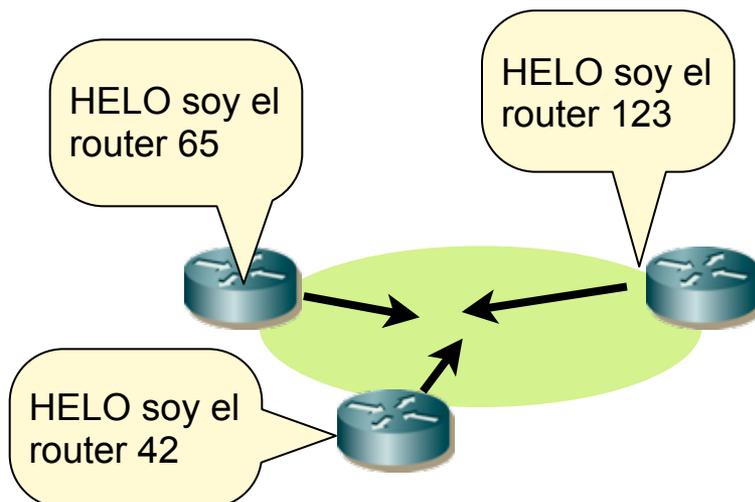
- Idea de funcionamiento
 - Cada router es responsable de reconocer a sus vecinos y aprenderse su nombre (identidad)
 - Cada router construye un paquete llamado LSP (link state packet) conteniendo una lista de todos sus vecinos y el coste asociado
 - El LSP se envía a todos los demás routers de la red. Cada router recuerda el ultimo LSP recibido de cada otro router
 - Cada router con el mapa completo de la topologia calcula las rutas a cada destino posible y actualiza su tabla de rutas
- Como de difícil es cada una de estas fases?

Link-State: 1 descubrir vecinos

- Descubrimiento de vecinos

Mensaje HELO

- Contiene una identidad del router
- Periódico a la red de área local o al enlace punto a punto
- Recordar los últimos HELOs vistos (lista de identidades de los vecinos)
- Tiempo entre mensajes
- Tiempo para dar al router por desaparecido



Vecinos	Visto hace
65	6s
42	14s

Link-State: construir LSPs

- Construir LSPs
 - Periódicamente
 - Cada vez que veo un nuevo vecino
 - Cada vez que cambia el coste a un vecino
 - Cada vez que dejo de ver a un vecino (ha pasado el tiempo definido sin escuchar HELOs de ese vecino)
- El LSP contiene

El router de identidad X tiene esta lista de vecinos

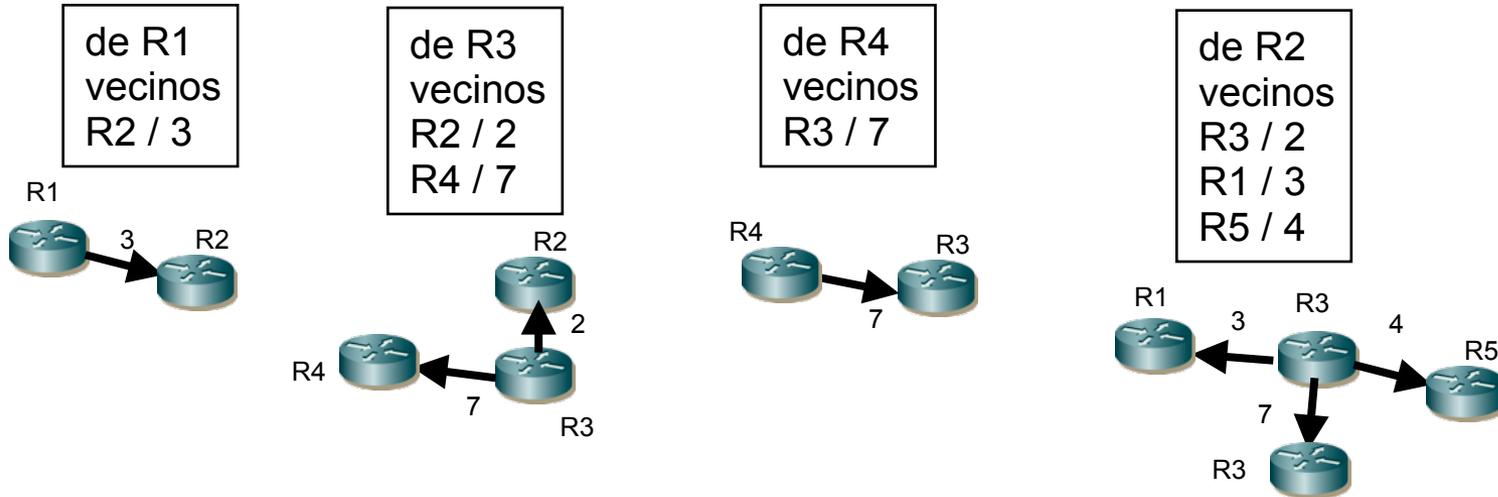
Vecino 1: Y con coste c1

Vecino 2: Z con coste c2

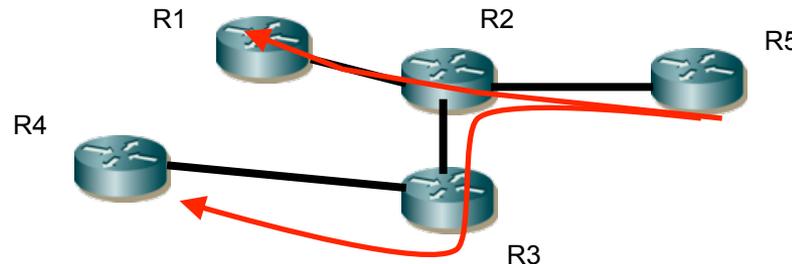
Vecino 3 ..
- En que se diferencia de los distance-vector??

Link-State: 3 distribuir LSPs

- Cada router necesita los LSPs de todos los demás, no solo el de sus vecinos
- Recibo estos



- Base de datos de nodos y enlaces en la red
- Usando el algoritmo de Dijkstra calculo caminos



Link-State: 3 distribuir LSPs

- Si hay un cambio en la red los routers que lo ven envían el cambio a todos
- El resto de los enlaces están en la base de datos
- Con el cambio de ese enlace se recalculan los caminos en todos
- No hay cuentas a infinito
- Reaccion muy rapida

Link-State: 3 distribuir LSPs

- El destino de los LSPs

Son todos los routers de la red

Es facil enviar a los vecinos pero no a cualquier router de la red... (no se las direcciones de todos, no puedo basarme en la tabla de rutas para llegar a ellos)

- Es necesario un sistema de envío a cualquier destino que no requiera que funcionen las tablas de rutas.

Inundación?

- La distribución de LSPs es el punto critico de los algoritmos de tipo link-state
 - Si no llega la información a todos, las rutas son incoherentes y puede haber ciclos de enrutamiento (recordar cuentas a infinito con un solo enlace en desacuerdo)
 - Si no tenemos cuidado la inundación puede causar mucho trafico extra, los routers pueden saturarse y dedicar gran parte de su cpu simplemente a recalcular rutas tras recibir

Link-State: 3 distribuir LSPs

- Inundando LSPs
- Cada router escucha LSPs de sus vecinos y reenvía
- Como limitar el tráfico
 - TTLs?
 - Con LSPs ya estamos guardando estado (cada router recuerda el ultimo LSP de cada destino) usar esto para limitar la inundacion (ignorar un LSP si ya lo he visto)

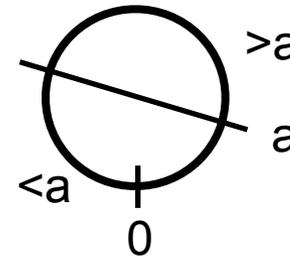
Sólo recordar el más reciente... no es tan facil

- No necesariamente llegan en orden
- Etiquetar los LSPs con la fecha (timestamp) de creacion?
 - Problemas si un nodo genera un timestamp en el futuro ya no se olvida y nunca se sustituye
 - Timestamp con significado global permite descartar pero es dificil tener tiempo global entre routers

Link-State: 3 distribuir LSPs

- LSP numeros de secuencia y edad
- Se asigna un numero de secuencia a cada LSP nuevo que se envía
 - Los routers solo se quedan con el LSP si tiene mas numero de secuencia que el anterior
- Casi... pero aun tiene problemas
 - Los campos tienen longitud finita... si llego al maximo vuelvo a 0

$a < b$ significa



- que pasa si un router se apaga y vuelve a encender y no recuerda el numero de secuencia que llevaba?
- que pasa si dos partes de la red quedan aisladas y mientras estan aisladas el numero de secuencia de un router avanza hasta el punto de que al volver a unirse los routers que llevan tiempo sin verlo piensan que sus numeros de secuencia son menores
- Se resuelve añadiendo edad a los LSPs

Un protocolo de distribución de LSPs

- Un router genera un LSP que contiene
 - La identidad del router origen
 - Un numero de secuencia que es 1 mas que el anterior LSP que envió
 - Una edad (tipo TTL) que es un valor maximo
 - La lista de vecinos con el peso
- Un router que recibe un LSP lo compara con el LSP que tiene almacenado para ese destino
 - Si tiene mayor numero de secuencia que el almacenado lo sustituye por el nuevo
 - Si el almacenado tiene edad 0 es sustituido independientemente de la secuencia (me olvido de los que llegan a edad 0)
 - Si he sustituido el LSP almacenado por el recibido lo propago enviandolo a todos los vecinos excepto al que me lo ha enviado
- Cada cierto tiempo descuento 1 a la edad de todos los LSPs almacenados
- Caso real: edad 3bits 0-7 x8s (56 segundos y se descuenta cada 8s)
se genera un LSP cada 60s (el primero al encender el router espera 90s)
- Parece razonable?

The ARPANET incident

- Ese algoritmo consiguió bloquear totalmente ARPANET
- Supongamos que un router funcionando mal envía tres LSPs seguidos con numeros de secuencia a, b, c de forma que $a < b < c$ pero $c < a$
- Con el algoritmo anterior como cada router que los recibe los acepta y reenvia porque cree que son nuevos, cada uno de los que lo recibe los acepta y reenvia y al recibirlos de nuevo los sigue aceptando.

Esto desencadena una inundación total, ocurrió en la realidad. Una noche ARPAnet dejo de funcionar. Al parar routers y examinarlos se veia que tienen la cola de salida saturada de esos tres LSPs toda ARPANET estaba al 100% enviando esos 3 mensajes

- El router que los había enviado tenía un fallo y fue desconectado

Pero eso no detuvo el problema todos los routers tenían copias de esos mensajes si reinicias un router en cuanto vuelve a funcionar sus vecinos le envían de nuevo copias y se llena

- Una solución obvia es apagar toda ARPANET
- Alguna otra forma de resolverlo?
- Hay que tener cuidado con los algoritmos distribuidos a gran escala

The ARPANET incident

- Qué falla en el protocolo?
 - Los mensajes no envejecen por que son sustituidos por nuevos antes de que pasen los 8s
 - Solución envejecer el mensaje en cuando un router lo toque

- Mejoras que se implementan hoy en dia
 - El numero de secuencia es lineal y no da la vuelta
 - El campo de edad se decrementa en 1 en cuanto un router lo recibe. Mientras esta almacenado se ira decrementando mas
 - Cuando se decide que un LSP es nuevo no se genera y se pone en cola sino que se marca que debe enviarse.

Quando el router decide que puede mandar un LSP elige uno que este marcado, lo construye y lo envia

De forma que nunca hay encolado mas de una copia del mismo LSP

 - Los LSP se confirman pero tampoco se encolan los ACKs solo un ACK pendiente por origen

- Básicamente el algoritmo anterior con estas modificaciones lo implementan todos los protocolos link-state modernos

Link-State: 4 calcular rutas

- Los LSPs se guardan en una link-state database
- Se utiliza el algoritmo de Dijkstra para calcular caminos
- Se colocan las rutas de este nodo teniendo en cuenta esos caminos
- Si la red es muy grande todos los routers necesitan mas recursos que en el caso de distance-vector
 - Deben almacenar todos los enlaces de la red
(mucha mas memoria que el vector de distancias)
 - Deben calcular el algoritmo de Dijkstra cada vez que ven un cambio
(mucho mas complicado que la iteracion del Bellman-Ford)
- La convergencia es rapida porque los LSPs llegan a todos y no hay ambigüedades de caminos ni cuentas a infinito.

Distance-vector vs Link-state

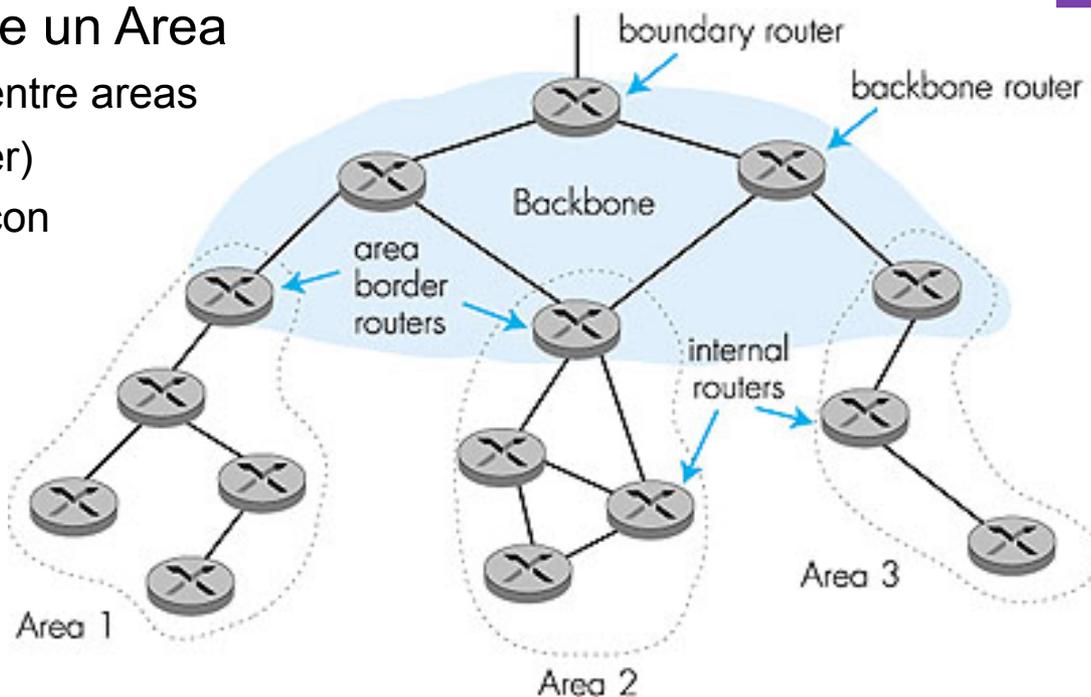
- En uso de memoria: menor distance-vector
- En tráfico de red: controvertido distance-vector no hace inundacion pero la inestabilidad dura mas
- En carga de CPU: menor distance-vector
- En robustez: no esta claro. link-state es mas robusto frente a fallos no maliciosos
- En funcionalidad: link-state tiene muchas ventajas, ya que todos los routers conocen la topologia entera, es mas facil de depurar las situaciones de fallo, permite source-routing, caminos paralelos...
- Velocidad de convergencia: calramente mejor link-state

Link-state en Internet

- OSPF (Open Shortest Path First) (RFC-2328)
 - Envío del estado de los enlaces entre routers
 - Mapa de la topología en cada router
 - Base de datos de estado de enlaces
 - Cálculo de las rutas con el algoritmo de Dijkstra en cada router
- Protocolo para enviar información sobre el estado de los enlaces
 - Link-state advertisements en la terminología de OSPF se llaman LSAs
 - Inundación (flooding) a todos los routers (del sistema usando OSPF)
 - Mensajes OSPF directamente sobre IP (no TCP ni UDP) protocol=89
 - Enrutamiento jerárquico para simplificar (areas)

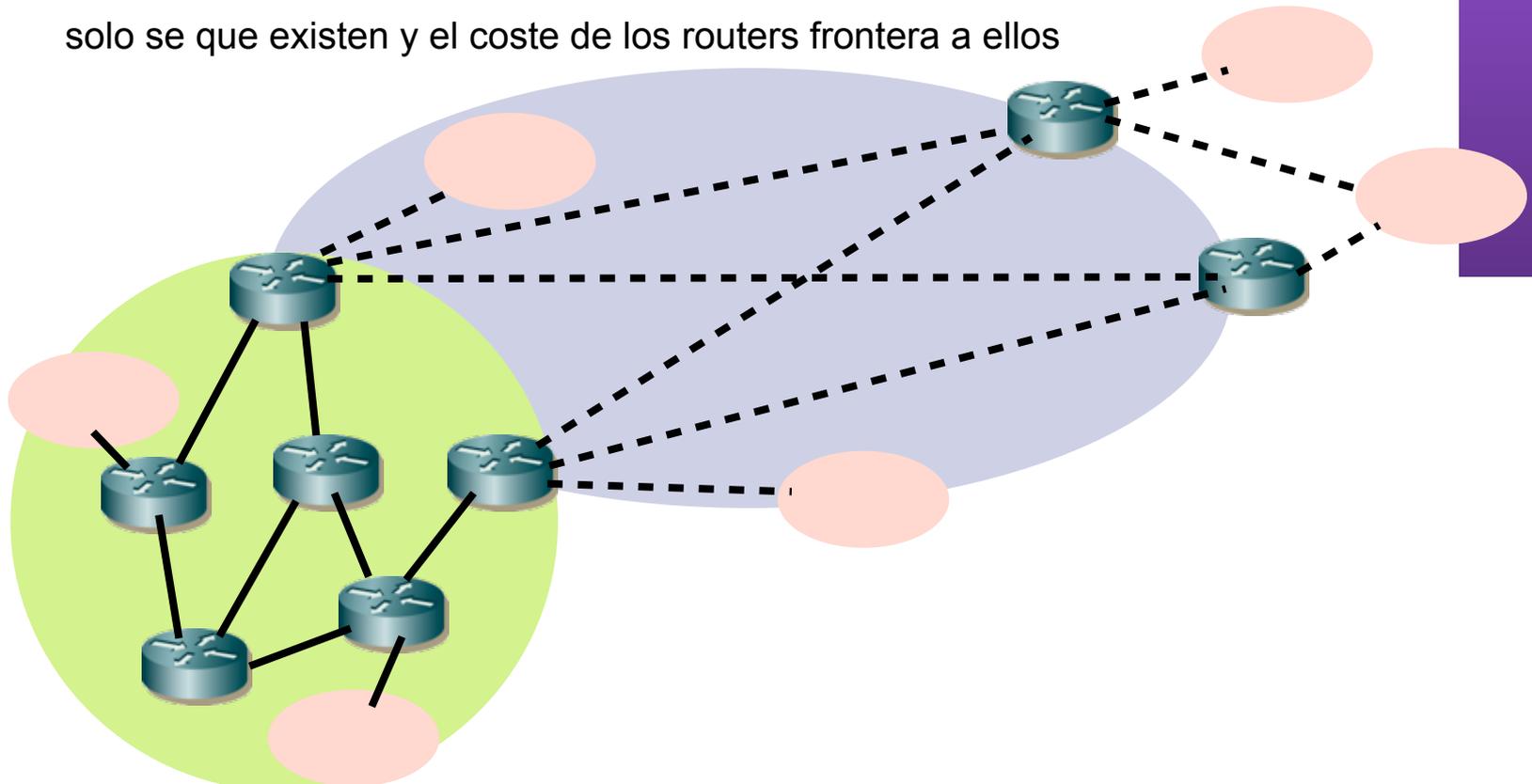
OSPF: áreas

- Enrutamiento jerárquico
 - El enrutamiento se calcula por **Áreas**
(cada nodo calcula la tabla de rutas en su área)
 - Área 0: backbone (enlaces a otras entidades)
 - Las áreas solo están conectadas al backbone no entre si
 - Routers en más de un Area
 - Routers frontera entre áreas
(border area router)
 - Routers frontera con otras entidades
(boundary router)



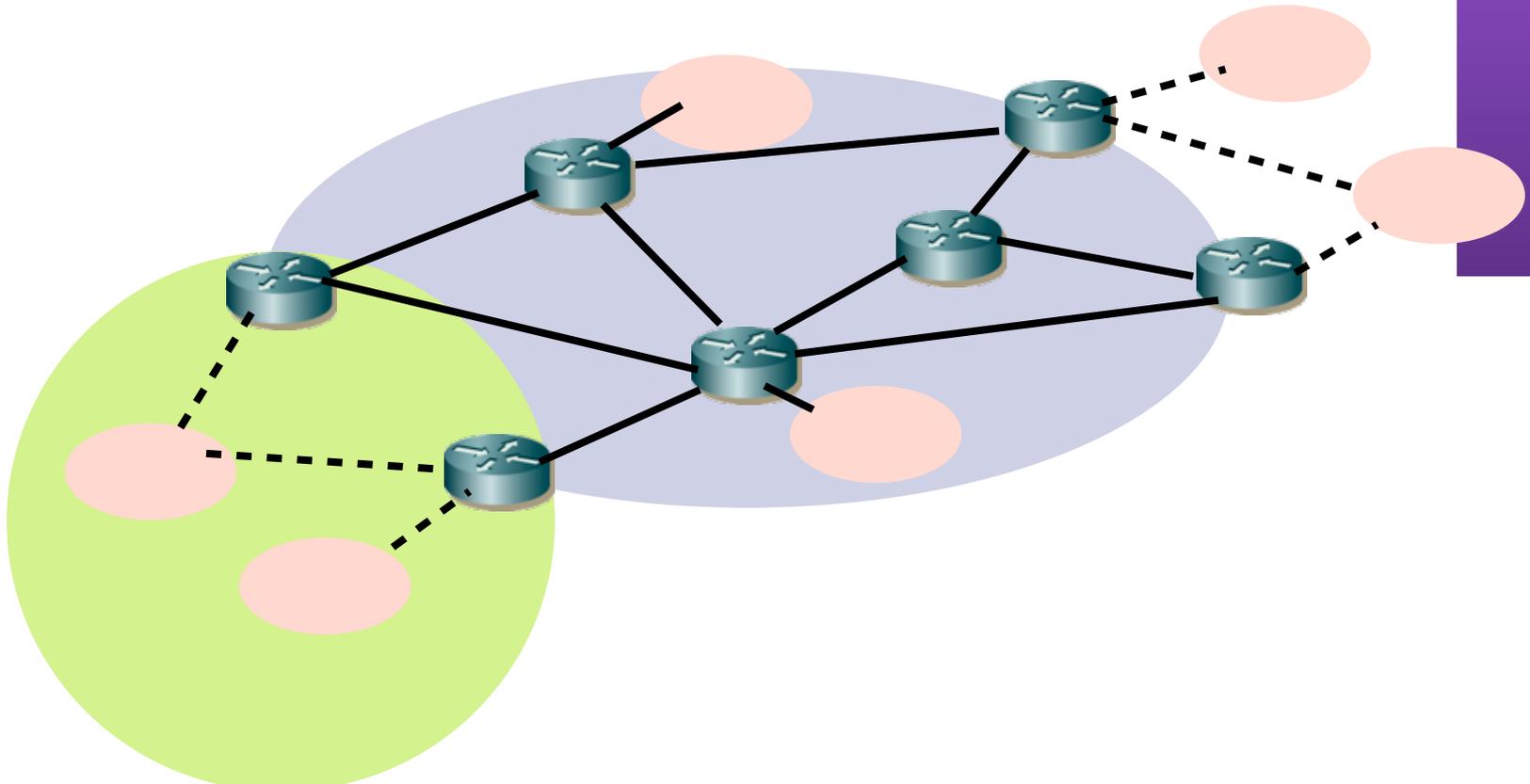
OSPF: áreas

- Los routers de un área deben conocer
 - Todos los enlaces internos a esa área
 - Todos los routers externos al área 0
 - Todos los destinos posibles en el area 0
 - Todos los destinos posibles en el exterior
- Cada área calcula un modelo simplificado del grafo con eso
 - Los enlaces externos no se si son enlaces directos solo se que existen y el coste de los routers frontera a ellos



OSPF: áreas

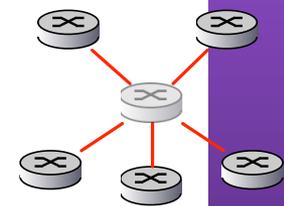
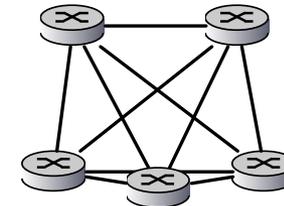
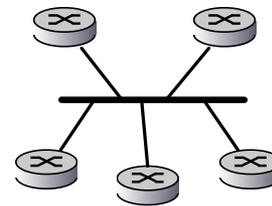
- Los routers del area 0 deben conocer
 - Todos los enlaces internos a esa área
 - Todos los destinos posibles en las demas áreas
 - Todos los destinos posibles en el exterior
- Su topología conocida es tambien simplificada



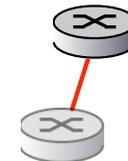
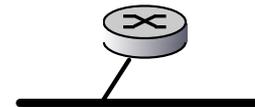
OSPF: enlaces

- Tipos de enlaces de los que informa OSPF

- Enlaces punto-a-punto
no necesitan dirección de red
- Transient link
subred con varios routers



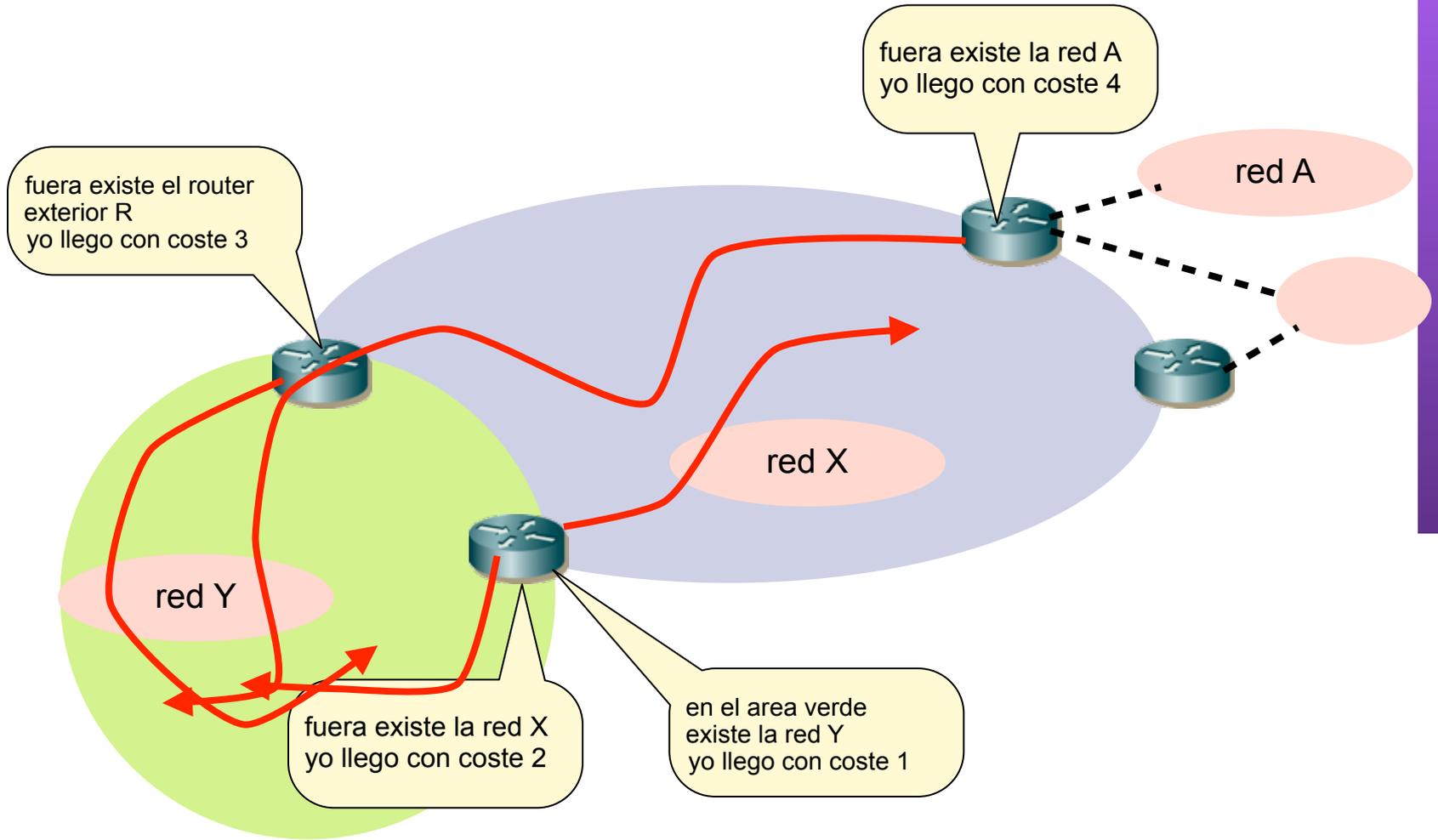
- Stub link
subred con un solo router
- Virtual link
a través de otros routers



OSPF: mensajes

- Los mensajes de OSPF envían información sobre enlaces
 - Router-link define los enlaces de un router
Se inunda a toda el área
 - Network-link define los enlaces de una red (transient link o stub link)
Para ello hay que elegir un designated router que mande estos mensajes.
Se inunda a toda el área
 - Summary-links envío de rutas conocidas por los routers de frontera
La generan los routers frontera para informar de lo que hay fuera o dentro de un area a los routers que no estan en ese área

OSPF: mensajes

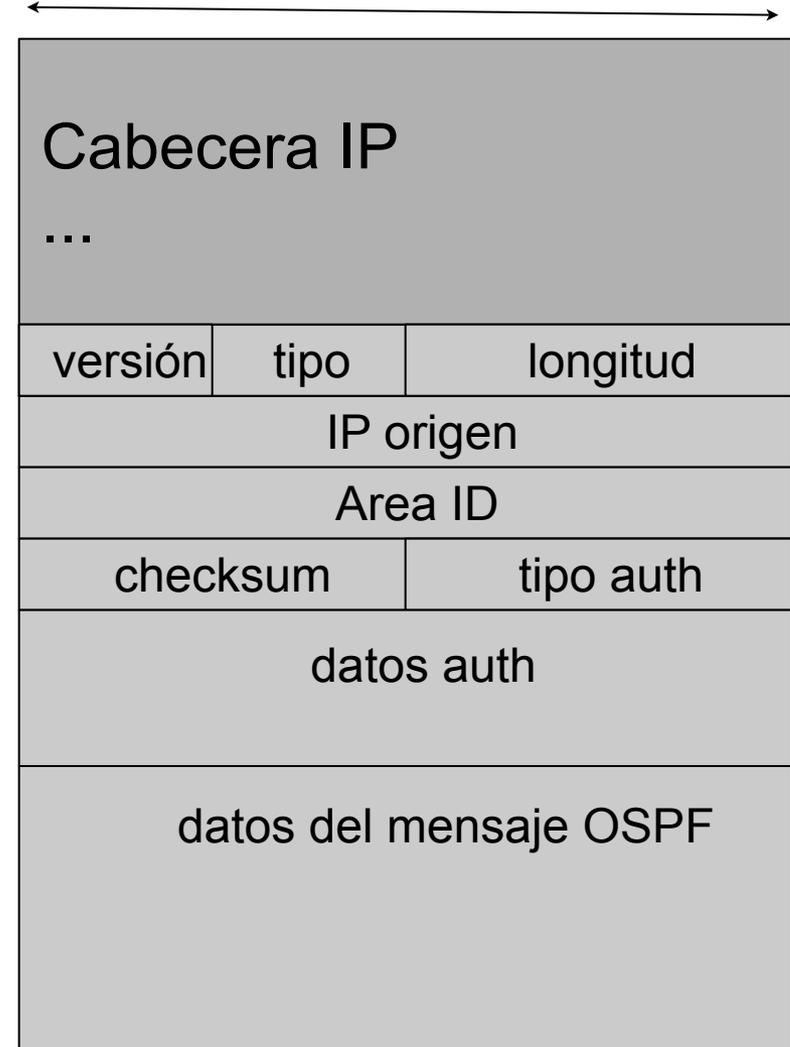


OSPF

32 bits

- Formato del paquete

- versión
- tipo de mensaje
 - 1 HELO
 - 2 DATABASE DESCRIPTION
 - 3 LINK STATE REQUEST
 - 4 LINK STATE UPDATE
 - 5 LINK STATE ACK
- longitud
- origen (identidad del que genero el LSA)
 - los paquetes se reenvian
- ID de area
- autenticación
 - tipo
 - datos
- Datos : un conjunto de LSAs



OSPF LSAs

- Tipos de LSAs

- Router link

- Informa de routers vecinos

- o redes stub vecinas

- Network link

- Informa de una red de area local con varios routers en el que se ha elegido un designated router. Solo el designated manda el LSA con una lista de todos los routers en esa red

- Summary link

- Informa de un destino red/mascara o router frontera existente en otra área

- AS external link

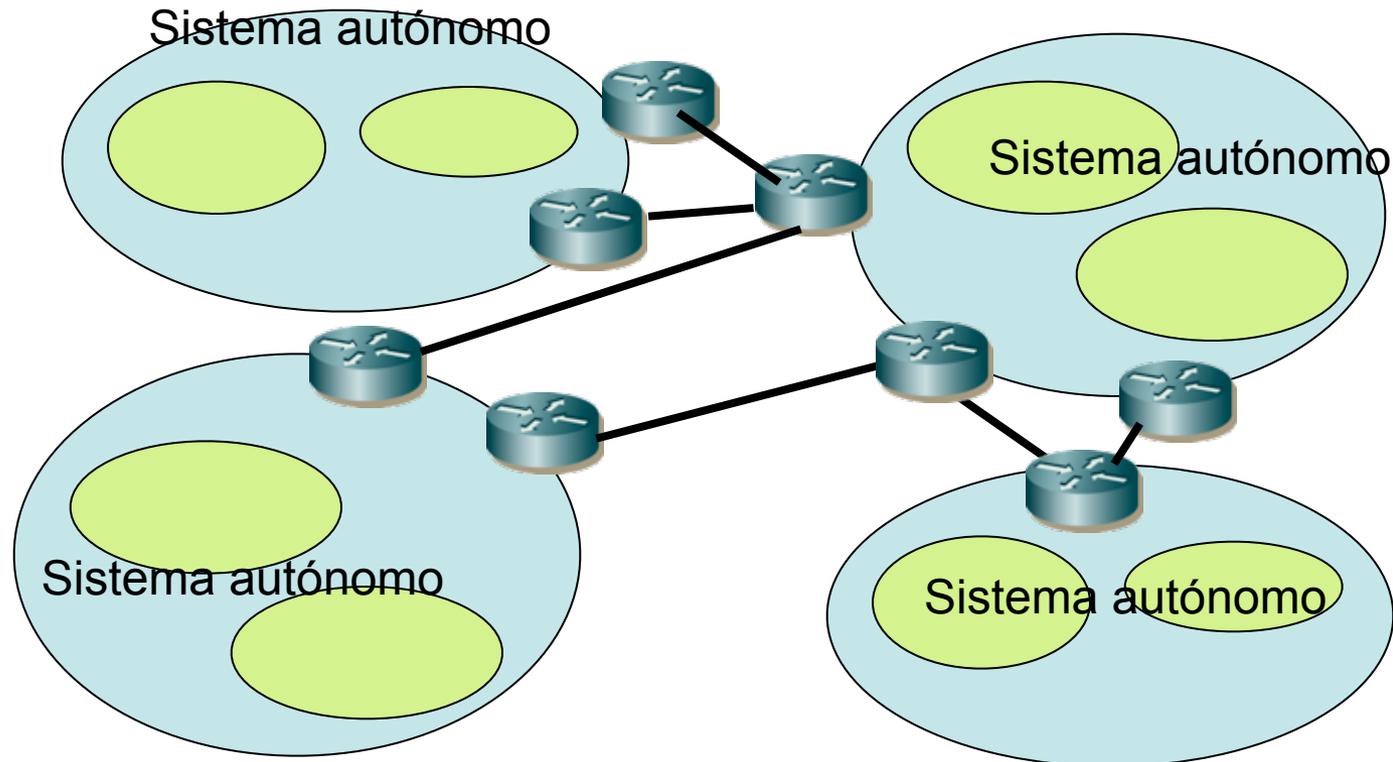
- Informa de un destino fuera del dominio OSPF

Otros protocolos link-state

- IS-IS
- Protocolos de enrutamiento de OSI
 - ES-IS: End System-to-Intermediate System
 - IDRP: InterDomain Routing Protocol
 - IS-IS: Intermediate System-to-Intermediate System
Originalmente para CLNP de OSI pero se extendió para permitir también IP
 - Es anterior a OSPF y muy parecido
- IS-IS es de tipo Link-state
 - Se usa sobre todo por ISPs
 - aunque viene del mundo OSI ha sido aceptado por el IETF
Soporta direcciones IP pero no estaba pensado para IP
 - Soporta enrutamiento jerárquico más complicado que OSPF
No solo un backbone-areas sino areas con más niveles de jerarquía
- Otros como PNNI (para ATM) o NLSP (Netware) soportan áreas con interconexión arbitraria...
 - Complejos y soluciones interesantes ... pero en desuso

Intradomain vs Interdomain

- Internet es un conjunto de sistemas autónomos con enrutamiento independiente
- Cada uno puede utilizar los protocolos de enrutamiento que elija internamente **intra-dominio**
- Hay protocolos para compartir las rutas entre dominios (**inter-dominio**)
 - Un protocolo común para hablar con cualquier otro sistema autónomo
 - Actualmente se usa BGP4



Interdomain routing

- Solo puede haber uno un protocolo unico BGP4
- No puede basarse en rutas por defecto y exterior
No hay exterior. Se le llama tambien default-free-zone
- Protocolo de intercambio de rutas entre sistemas autonomos vecinos
 - Los AS son los nodos del grafo
 - El protocolo debe soportar cuestiones politicas aparte de pesos. (i.e. el trafico que va de un destino a otro puede ir por este AS y no por este otro porque uno es cliente mio y no puedo usarlo como transporte)
 - BGP4 es de tipo PATH-Vector (Bellman-Ford pero anuncio el camino completo no solo el peso)

Conclusiones

- Protocolos de enrutamiento link-state
 - Mas complejos
 - pero en general mejores que distance-vector
- Protocolos link-state reales: OSPF (+IS-IS +otros)
- Usan enrutamiento jerarquico
- Enrutamiento intradominio e interdominio