

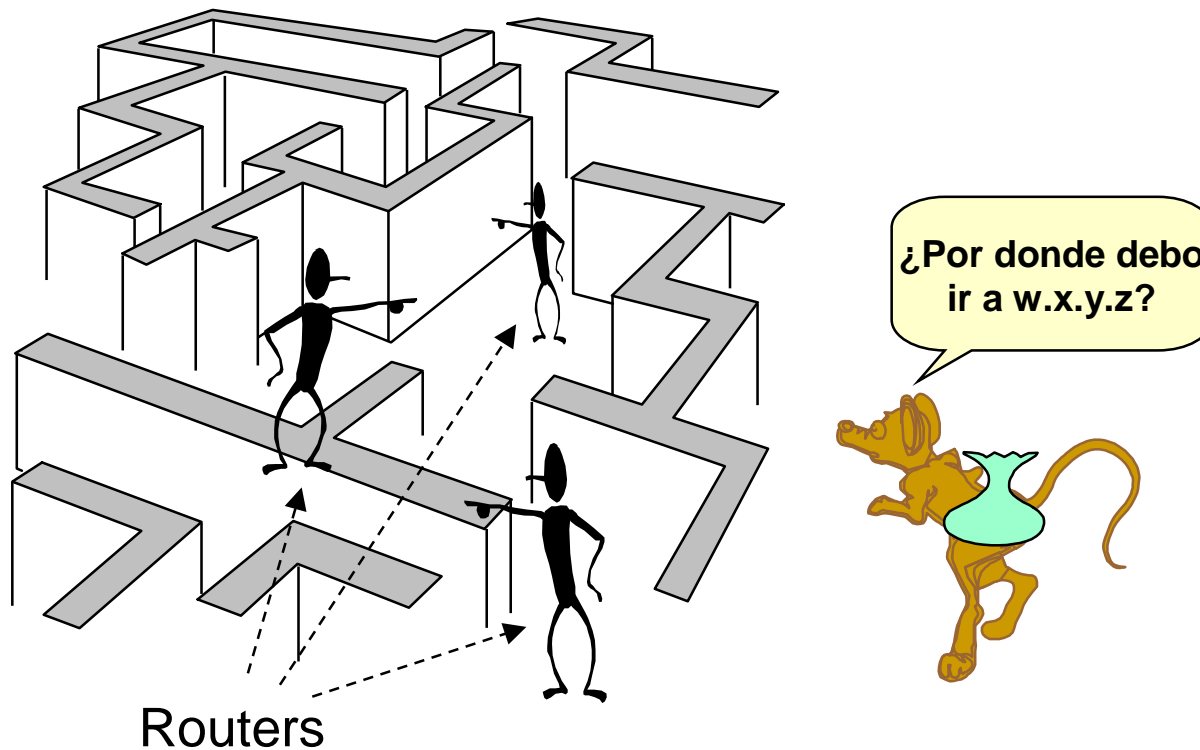
Capítulo 6

El Nivel de Red: Generalidades

Sumario

- **Aspectos generales del nivel de red**
- Algoritmos de routing
- Control de congestión

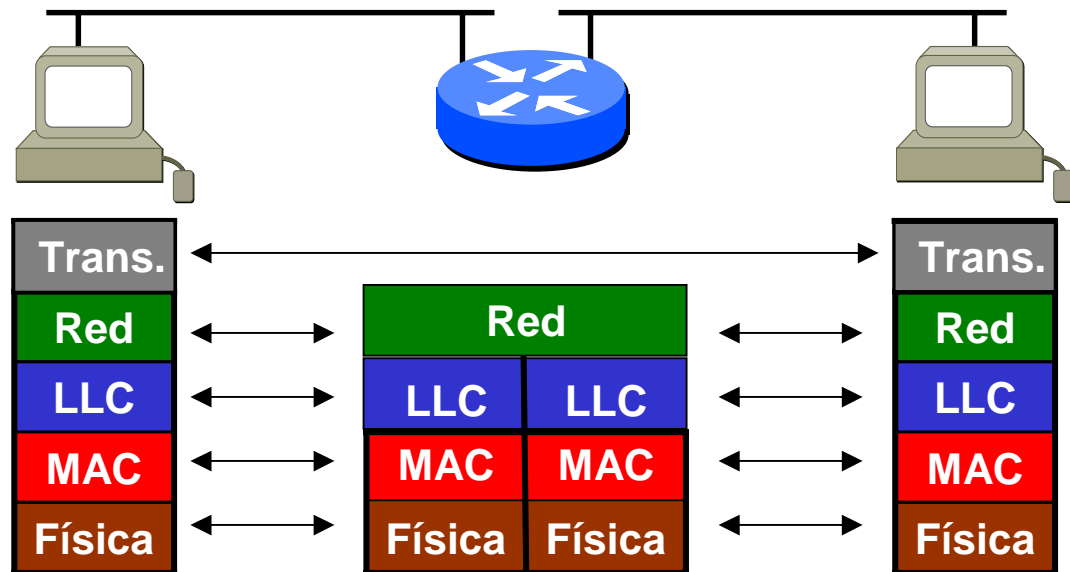
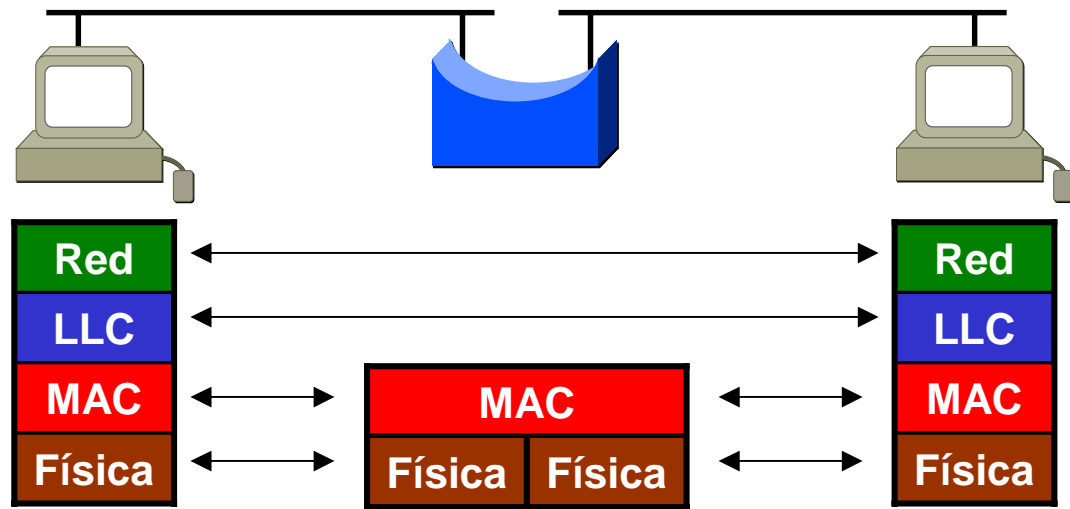
La Capa de Red



El nivel de Red

- Es la capa por antonomasia, la única que ‘ve’ la red.
- Se constituye con dos tipos de nodos:
 - Nodos terminales (Hosts)
 - Nodos de tránsito (Routers o Conmutadores)
- Normalmente los routers tienen varias interfaces y los hosts una (pero puede haber hosts ‘multihomed’).
- Los routers y las líneas que los unen constituyen la subred, gestionada por el proveedor u operador.
- En una comunicación LAN-LAN el nivel de red es casi inexistente (no hay ‘nodos de tránsito’).

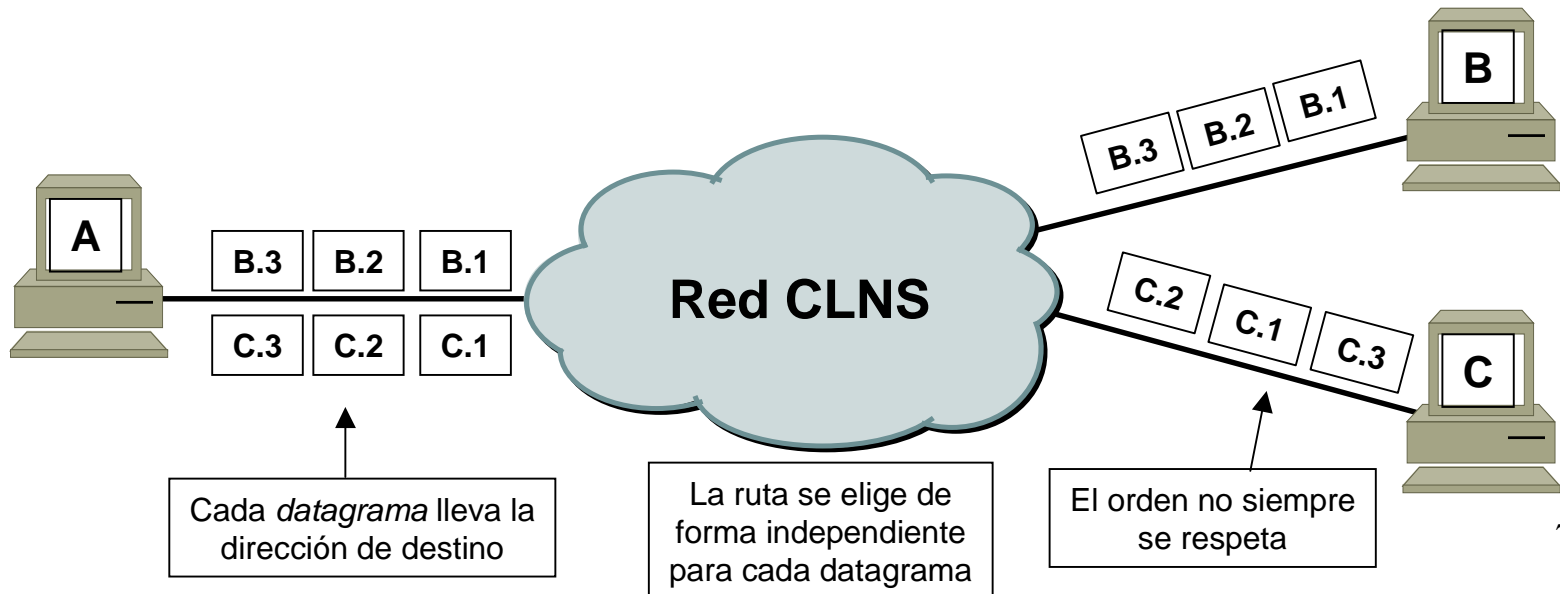
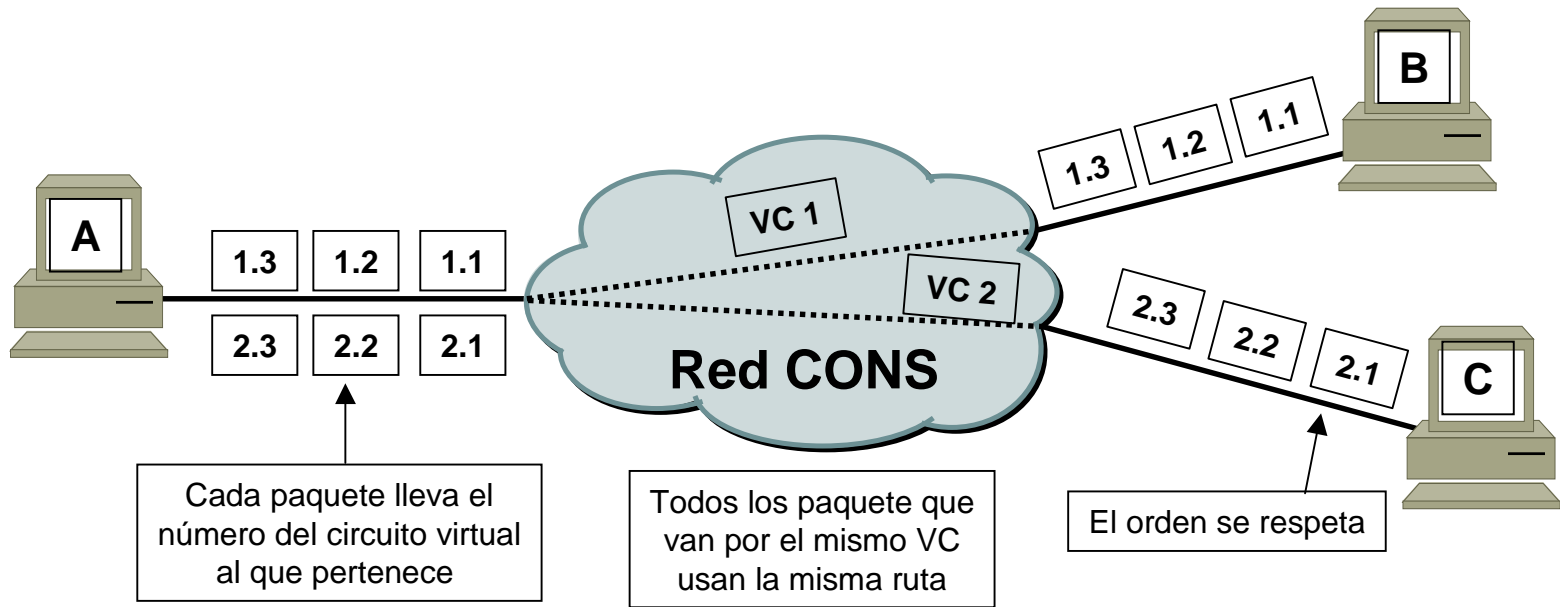
Comunicación mediante un puente y un router



Funciones del nivel de Red

- Elegir la ruta óptima de los paquetes
 - Servicio CONS: solo en el momento de establecer el VC
 - Servicio CLNS: para cada datagrama enviado
- Controlar y evitar la congestión
- Controlar que el usuario no abuse del servicio (excede lo pactado)
- Resolver (‘mapear’) las direcciones de nivel de red con las de nivel de enlace (p. Ej. en LANs).

CONS vs CLNS



	Red CLNS (datagramas)	RED CONS (circuitos virtuales)
Establecimiento conexión	Innecesario	Requerido (permanente o temporal)
Direccionamiento	Cada paquete lleva la dirección completa de origen y destino	Los paquetes solo llevan el número del VC (generalmente pequeño)
Información de estado	La subred no conserva ninguna	Cada VC requiere una entrada en las tablas de cada conmutador
Routing	Independiente para cada paquete	La ruta se elige al establecer el VC; todos los paquetes siguen esa ruta
Efecto de fallo en un router	Se pierden paquetes en tránsito solamente	Todos los VC que pasan por ese router se terminan
Control de congestión	Difícil	Mas fácil

Sumario

- Aspectos generales del nivel de red
- **Algoritmos de routing**
- Control de congestión

Algoritmos de routing

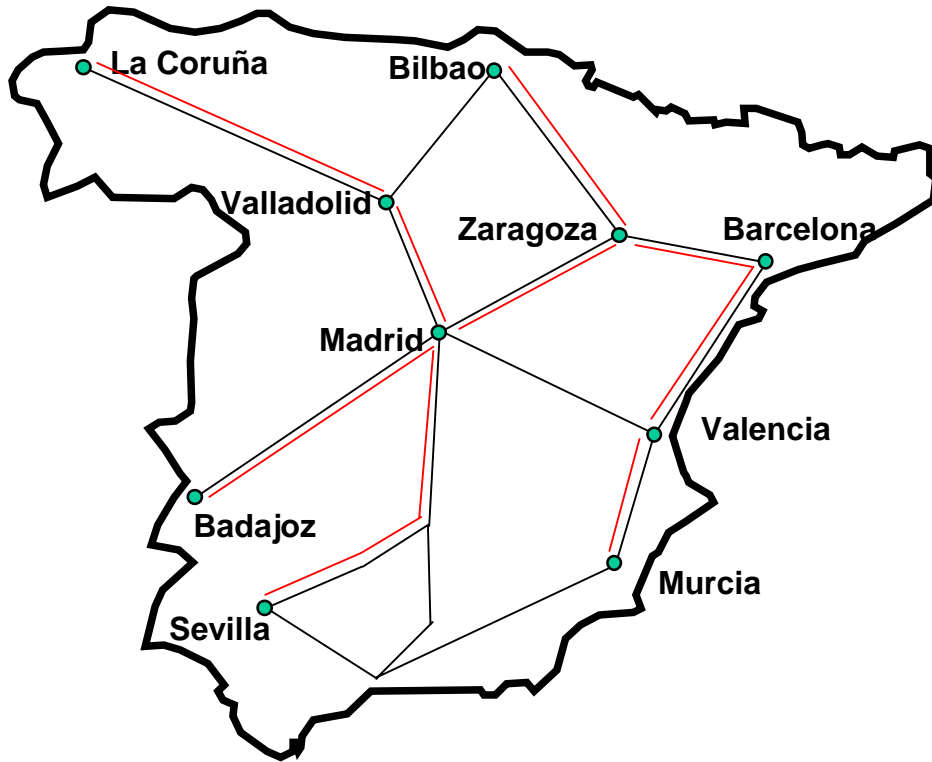
- Los algoritmos de routing pueden ser:
 - Estáticos: toman decisiones en base a información recopilada con anterioridad. La ruta no cambia.
 - Dinámicos: deciden en base a información obtenida en tiempo real. Requieren un protocolo de routing para recoger la información. La ruta puede cambiar constantemente.
- Salvo en redes muy simples o en zonas periféricas casi siempre se utiliza routing dinámico.

Principio de optimalidad

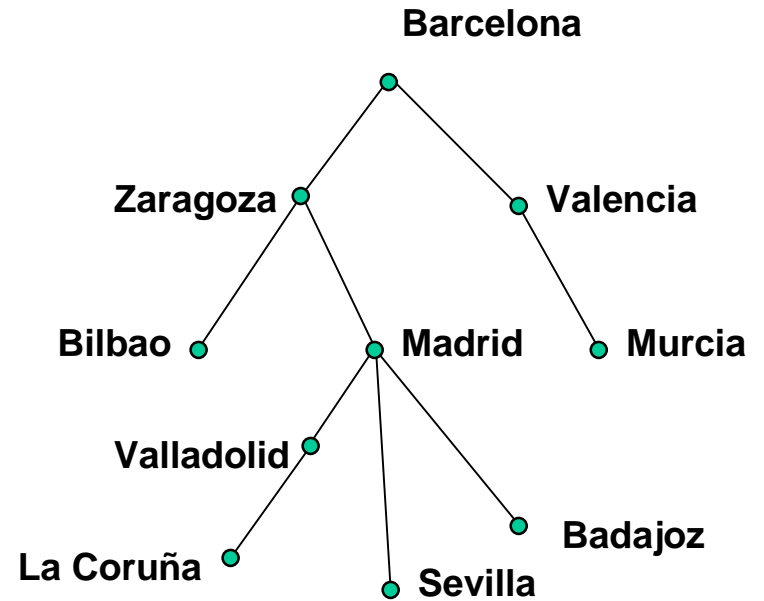
Si Valencia está en la ruta óptima de Murcia a Barcelona, entonces el camino óptimo de Valencia a Barcelona está incluido en la ruta óptima de Murcia a Barcelona

Corolario: Todas las rutas óptimas para llegar a Barcelona desde cualquier sitio forman un árbol sin bucles (spanning tree) con raíz en Barcelona.

Principio de optimalidad



La red de autopistas
españolas



Rutas óptimas hacia
Barcelona

Concepto de ruta óptima en viajes por carretera

- Para elegir la ruta óptima se pueden aplicar diversos criterios, por ejemplo:
 - La que minimice la distancia
 - La que minimice el tiempo
 - La que minimice el consumo de gasolina
 - La que minimice el costo (p. ej. evitar peajes)
 - La que minimice el cansancio (preferible autopistas, pocas curvas, cambios de carretera, etc.)
 - Una determinada combinación de todos los anteriores con diversos pesos según los gustos del usuario
- La ruta óptima puede variar en función del criterio elegido. Ver por ejemplo www.michelin.com

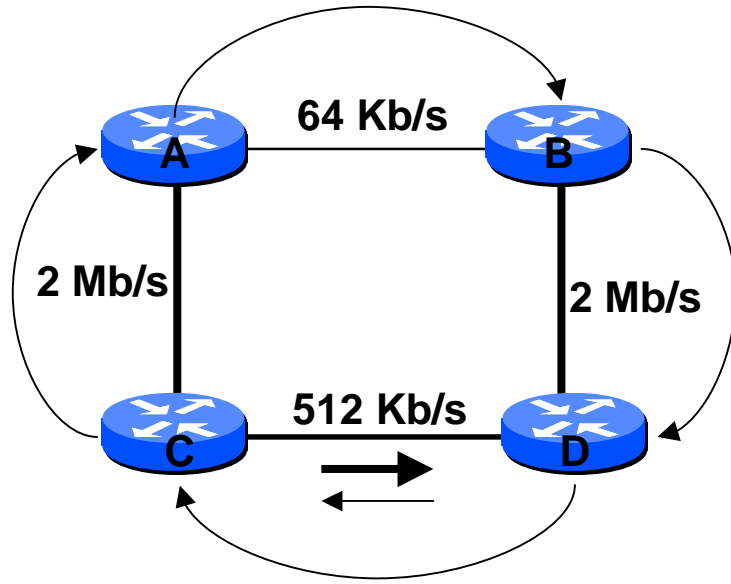
Concepto de ruta óptima en telemática

- Los criterios que se aplican suelen ser:
 - Minimizar el número de routers (saltos) por lo que se pasa
 - Maximizar el caudal (ancho de banda) de los enlaces por los que se pasa
 - Minimizar el nivel de ocupación o saturación de los enlaces que se atraviesan
 - Minimizar el retardo de los enlaces
 - Maximizar la fiabilidad de los enlaces (minimizar la tasa de errores)
 - Una determinada combinación de todos los anteriores con diversos pesos según los gustos del usuario

Encaminamiento estático basado en el flujo

- Consiste en optimizar las rutas de acuerdo con la capacidad y el nivel de ocupación de cada enlace, calculado a partir del tráfico medio previsto entre nodos.
- Se plantean varias topologías, se comparan todas y se elige la más adecuada.
- Es preciso disponer de información que permita estimar el tráfico medio entre cada par de nodos.
- Interesante para decidir la topología cuando se diseña una red de líneas punto a punto (Por ej.: Frame Relay o ATM)
- No permite responder con rapidez a cambios en la red.

Ejemplo de routing estático



Matriz de tráfico (MB/día)

	Destino			
	A	B	C	D
	A	-	10	0
	B	10	-	0
	C	0	0	-
	D	0	0	-

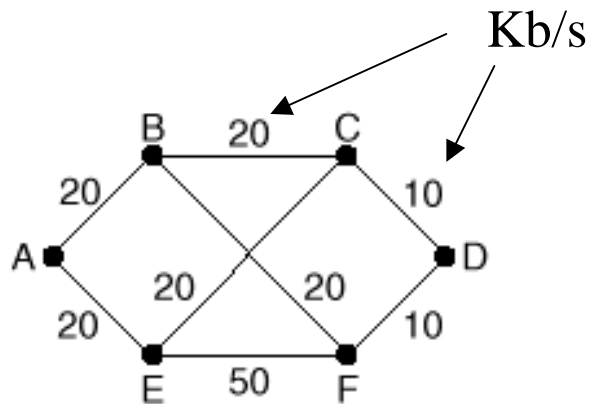
Origen

El enlace C-D está muy saturado (91%) en el sentido C→D pero es poco utilizado (4%) en sentido D→C

Ruta óptima de A a B: A → B

Ruta óptima de B a A: B → D → C → A

Ejemplo encaminamiento estático basado en el flujo



(a)

Paq/s de A a F

Ruta de A a F

(se puede modificar)

		Destination					
		A	B	C	D	E	F
Source	A		9 AB	4 ABC	1 ABFD	7 AE	4 AEF
	B	9 BA		8 BC	3 BFD	2 BFE	4 BF
	C	4 CBA	8 CB		3 CD	3 CE	2 CEF
	D	1 DFBA	3 DFB	3 DC		3 DCE	4 DF
	E	7 EA	2 EFB	3 EC	3 ECD		5 EF
	F	4 FEA	4 FB	2 FEC	4 FD	5 FE	

(b)

Fig. 5-8. (a) A subnet with line capacities shown in kbps. (b) The traffic in packets/sec and the routing matrix.

Tiempo de servicio (T): Tiempo medio que tarda en enviarse un paquete (tiempo de transmisión + tiempo en espera).

Por teoría de colas:

$T = 1 / (\mu c - \lambda)$, donde:

$\mu = 1 / \text{tamaño-paquete}$

c : velocidad

λ : Caudal medio (paquetes/seg)

Ej.: paq. 800 bits ($\mu = 0,00125$), $c = 20 \text{ Kb/s}$:

$\lambda = 0 \text{ paq/seg.}:$ $T = 40 \text{ ms}$

$\lambda = 1 \text{ paq/seg:}$ $T = 42 \text{ ms}$

$\lambda = 14 \text{ paq/seg:}$ $T = 91 \text{ ms}$

$\lambda = 24 \text{ paq/seg:}$ $T = 1000 \text{ ms}$

$\lambda = 25 \text{ paq/seg}$ $T = \infty$

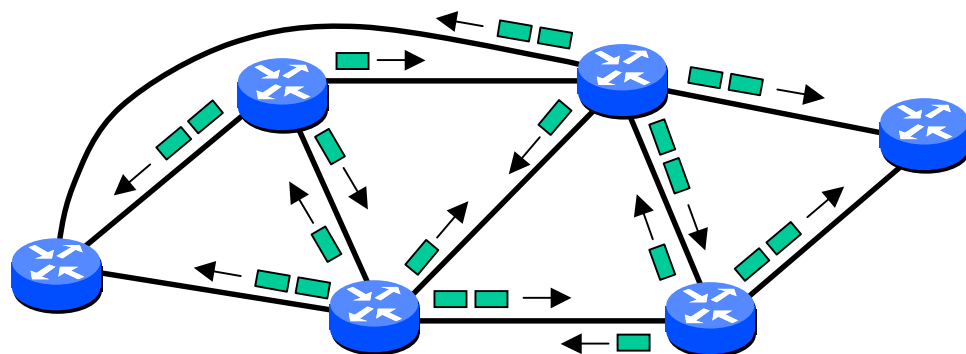
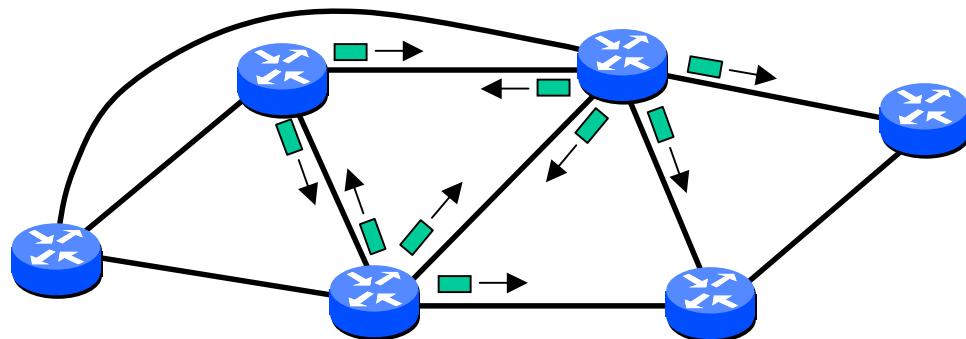
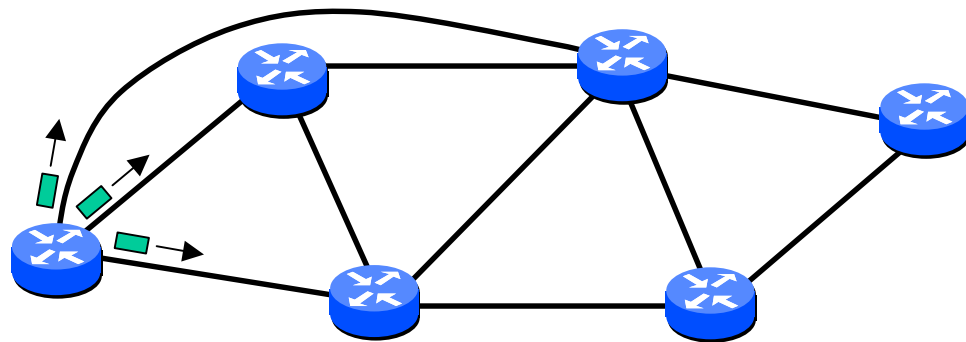
Cálculos de rutas con encaminamiento estático

		Caudal medio	Veloc.	Caudal max.	Tiempo Servicio	Caudal Medio Normaliz. ($\lambda_i / \sum \lambda_i$)
i	Line	λ_i (pkts/sec)	C_i (kbps)	μC_i (pkts/sec)	T_i (msec)	Weight
1	AB	14	20	25	91	0.171
2	BC	12	20	25	77	0.146
3	CD	6	10	12.5	154	0.073
4	AE	11	20	25	71	0.134
5	EF	13	50	62.5	20	0.159
6	FD	8	10	12.5	222	0.098
7	BF	10	20	25	67	0.122
8	EC	8	20	25	59	0.098

Fig. 5-9. Analysis of the subnet of Fig. 5-0 using a mean packet size of 800 bits. The reverse traffic (*BA*, *CB*, etc.) is the same as the forward traffic.

Encaminamiento por inundación

- Consiste en enviar cada paquete por todas las interfaces, excepto por la que ha llegado. Se utiliza en algunos algoritmos de routing multicast.
- Se utilizaba en los puentes transparentes.
- Si hay bucles se envían duplicados y el tráfico se multiplica. Para evitarlo se suele limitar el número de saltos.
- Otra posibilidad es que cada router mantenga una lista de paquetes enviados y descarte duplicados.
- La *inundación selectiva* envía solo por las líneas que aproximadamente van en la dirección correcta.



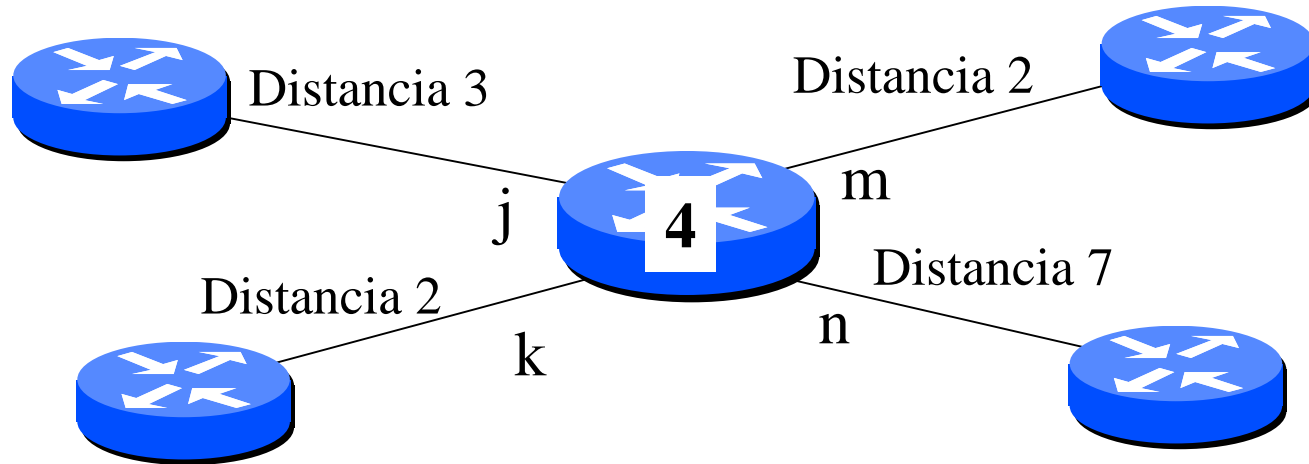
Transmisión
broadcast por
inundación con
límite de 3 saltos.

Encaminamiento dinámico

- Requiere recabar información en tiempo real sobre el estado de los enlaces
- Permite responder a situaciones cambiantes, p. Ej.: fallo o saturación de un enlace (solo si hay ruta alternativa).
- Dos algoritmos:
 - Vector distancia
 - Estado del enlace
- En ambos casos el cálculo de rutas óptimas se realiza de forma distribuida en toda la red.

Algoritmo del vector distancia (o de Bellman-Ford)

- Cada router conoce:
 - Su identificador
 - Sus interfaces
 - La distancia hasta el siguiente router de cada interfaz
- Cada router construye su base de datos de destinos, que indica por que interfaz enviar los paquetes para cada destino.
- Para esto los routers intercambian vectores de distancia, que indican la distancia a cada destino



Destino:

1 2 3 4 5 6 7 8 9 10 11

Recibido de j (+3)

12	3	15	3	12	5	6	18	0	7	15
----	---	----	---	----	---	---	----	---	---	----

Recibido de k (+2)

5	8	3	2	10	7	4	20	5	0	15
---	---	---	---	----	---	---	----	---	---	----

Recibido de m (+2)

0	5	3	2	19	9	5	22	2	4	7
---	---	---	---	----	---	---	----	---	---	---

Recibido de n (+7)

6	2	0	7	8	5	8	12	11	3	2
---	---	---	---	---	---	---	----	----	---	---

Vector calculado:

2	6	5	0	12	8	6	19	3	2	9
---	---	---	---	----	---	---	----	---	---	---

Tabla de rutas:

m	j	m	0	k	j	k	n	j	k	n
---	---	---	---	---	---	---	---	---	---	---

El problema de la cuenta a infinito



Distancia hacia A:

A apagado:	-	∞	∞
------------	---	----------	----------

A se enciende:	0	∞	∞
----------------	---	----------	----------

	0	1	∞
--	---	---	----------

	0	1	2
--	---	---	---

A se apaga:	-	3	2
-------------	---	---	---

	-	3	4
--	---	---	---

	-	5	4
--	---	---	---

	-	5	6
--	---	---	---

	-	7	6
--	---	---	---

	-	7	8
--	---	---	---

	-	9	8
--	---	---	---

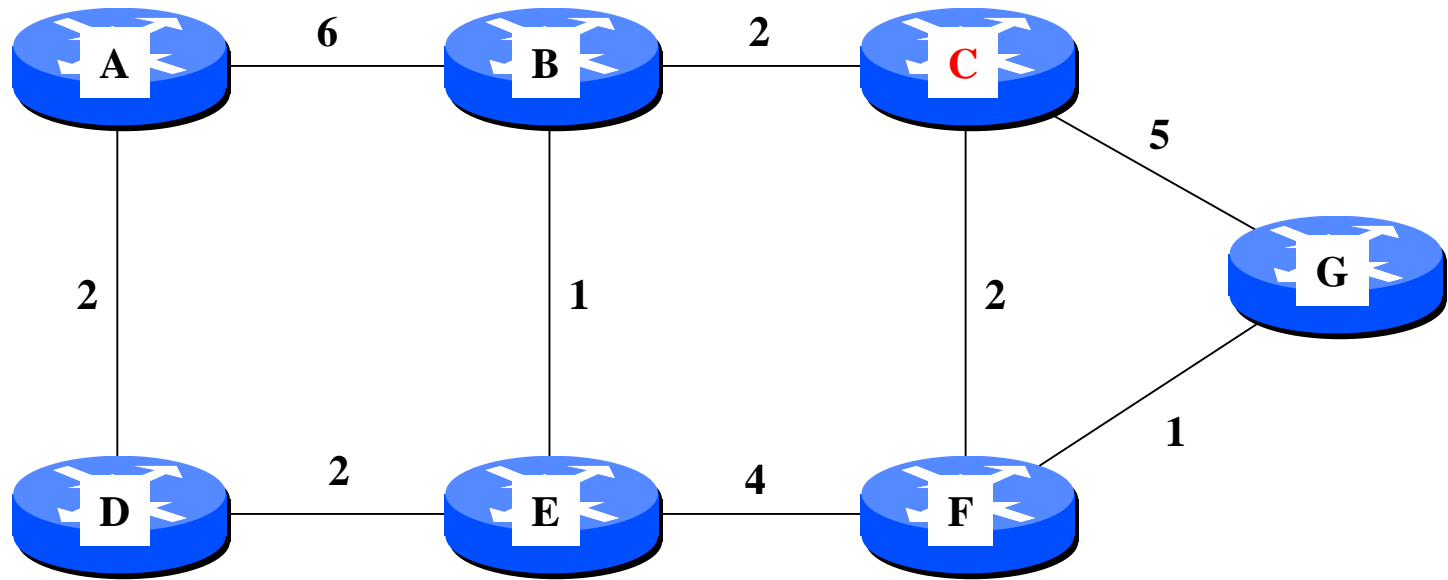
.....

Problema de cuenta a infinito

- Las noticias buenas viajan deprisa, las malas despacio.
- Hay diversos ‘trucos’ para evitar el problema de la cuenta a infinito, pero ninguno infalible.
- El vector distancia se utiliza actualmente en diversos protocolos de routing:
 - Internet: RIP, BGP, IGRP, EIGRP
 - También en Appletalk y versiones antiguas de DECNET e IPX

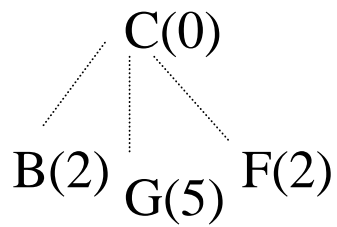
Algoritmo del estado del enlace

- Cada router contacta con sus vecinos y mide su ‘distancia’ a ellos.
- Construye un paquete LSP (Link State Packet) que dice:
 - Quién es él
 - La lista de sus vecinos y sus distancias a ellos
- Envía su LSP por inundación a todos los routers de la red
- Recaba los LSPs de todos los demás nodos
- Calcula las rutas óptimas por el algoritmo de Dijkstra:
 - Se pone él mismo como raíz del árbol, y coloca a sus vecinos
 - Mira los LSP de sus vecinos y despliega el árbol; cuando aparece más de un camino hacia un nodo se coge el más corto y se descartan los demás.
 - Las ramas son en principio provisionales. Una rama se confirma cuando es más corta que todas las demás provisionales.

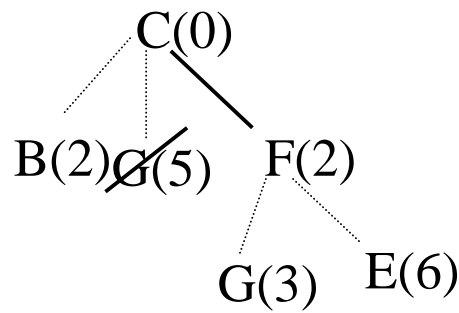


Link
State
Packets

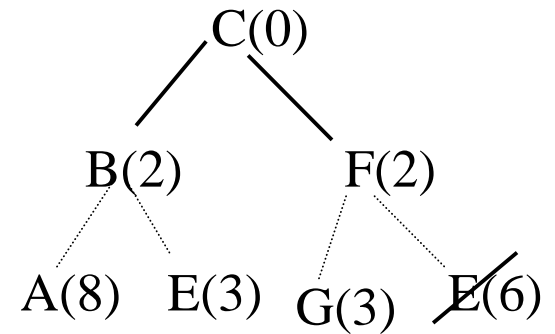
A	B	C	D	E	F	G
B/6	A/6	B/2	A/2	B/1	C/2	C/5
D/2	C/2	F/2	E/2	D/2	E/4	F/1
	E/1	G/5		F/4	G/1	



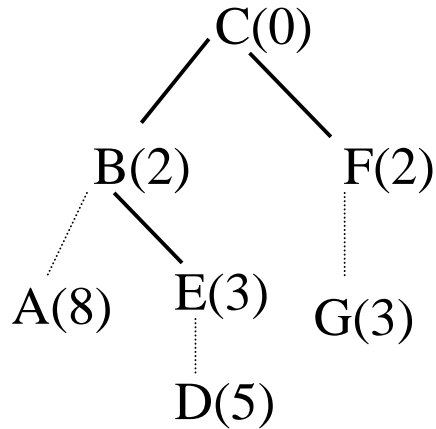
Coloca C en el árbol.
Examina el LSP de C



Coloca F en el árbol.
Examina el LSP de F.
Encontrado mejor camino a G

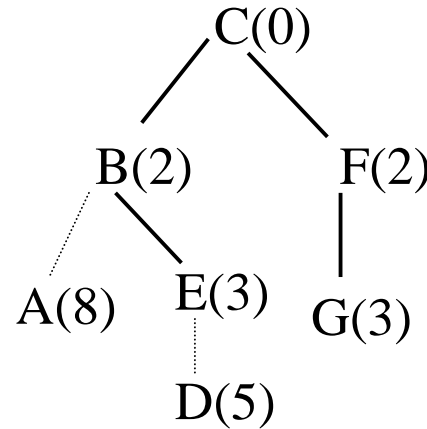


Coloca B en el árbol.
Examina el LSP de B.
Encontrado mejor camino a E

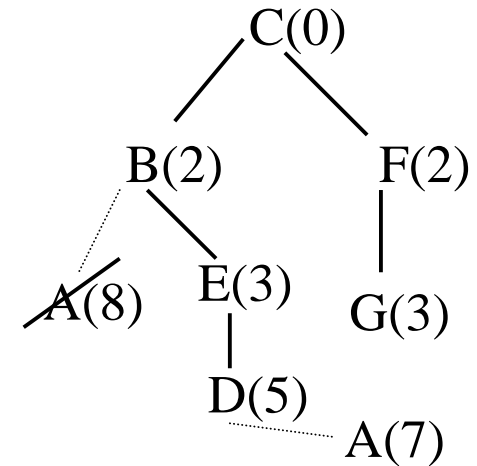


Coloca E en el árbol.
Examina el LSP de E.

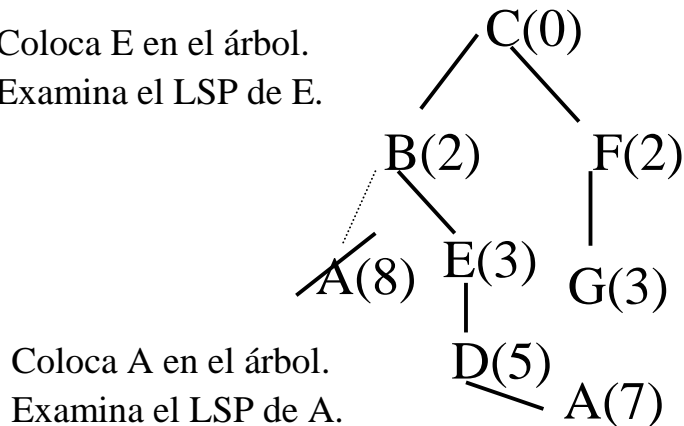
Algoritmo de Dijkstra



Coloca G en el árbol.
Examina el LSP de G.



Coloca D en el árbol.
Examina el LSP de G.

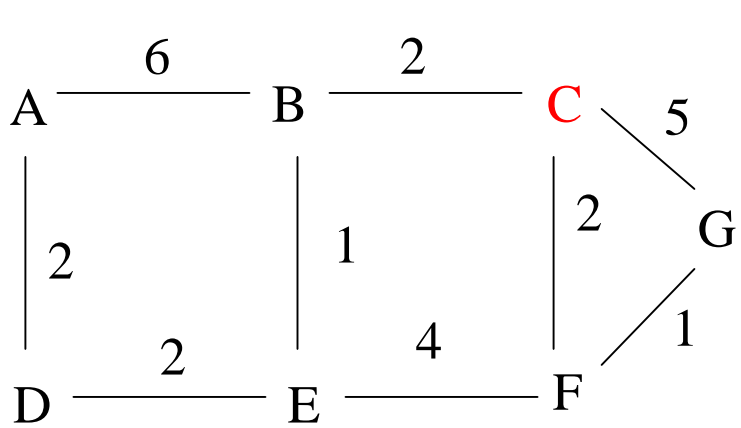


Coloca A en el árbol.
Examina el LSP de A.
No quedan nodos. terminar

A	B	C	D	E	F	G
B/6	A/6	B/2	A/2	B/1	C/2	C/5
D/2	C/2	F/2	E/2	D/2	E/4	F/1
	E/1	G/5		F/4	G/1	

Optimizaciones del algoritmo de estado del enlace

- Los LSP se numeran para detectar y descartar duplicados. Además tienen un tiempo de vida limitado.
- La inundación se hace reenviando cada LSP por todas las interfaces excepto por la que se recibió.
- Para evitar bucles solo se envían los LSP que son nuevos (nuevo = no estaba en la base de datos) y no están expirados.
- Con routing por el estado del enlace cada nodo tiene el 'mapa' detallado de toda la red (no ocurría con vector distancia).



LSPs

A	B	C	D	E	F	G
B/6	A/6	B/2	A/2	B/1	C/2	C/5
D/2	C/2	F/2	E/2	D/2	E/4	F/1
	E/1	G/5		F/4	G/1	

Flags
enviar
LSP

Flags
enviar
ACK

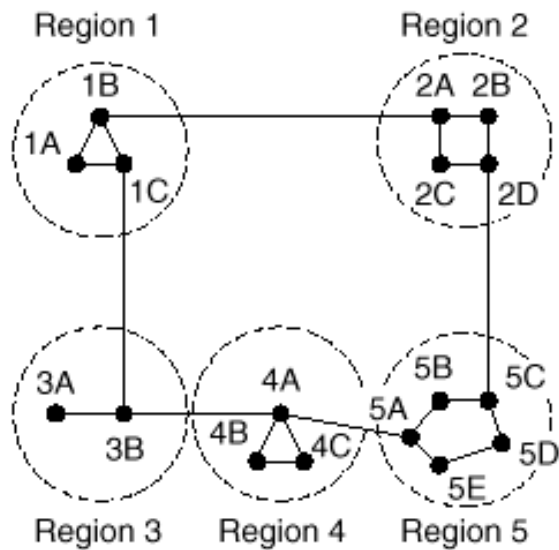
Origen	Secuen.	Edad	B	F	G	B	F	G	Datos
A	21	60	0	1	1	1	0	0	B/6, D/2
B	21	60	0	1	1	1	0	0	A/6, C/2, E/1
D	21	60	0	1	1	1	0	0	A/2, E/2
E	20	58	0	1	1	1	0	0	B/1, D/2, F/4
F	21	59	1	0	1	0	1	0	C/2, E/4, G/1
G	21	62	1	0	1	0	1	0	C/5, F/1
C	21	61	1	1	1	0	0	0	B/2, F/2, G/5

Routing por estado del enlace

- Generalmente se considera que los algoritmos del estado del enlace son mas fiables y eficientes que los del vector distancia.
- Se utiliza en diversos protocolos de routing:
 - Internet: OSPF, IS-IS
 - ATM: PNNI
 - DECNET

Routing jerárquico

- **Problema:** los algoritmos de routing no son escalables. La información intercambiada aumenta de forma no lineal con el tamaño de la red. Lo mismo ocurre con la complejidad de los cálculos.
- **Solución:** crear niveles jerárquicos. Solo algunos routers de cada región comunican con el exterior. Las rutas son menos óptimas pero se reduce la información de routing.
- Parecido a la forma como se organizan las rutas en la red de carreteras (internacionales, nacionales, regionales).



(a)

Full table for 1A

Dest.	Line	Hops
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	1C	5
5C	1B	5
5D	1C	6
5E	1C	5

(b)

Hierarchical table for 1A

Dest.	Line	Hops
1A	—	—
1B	1B	1
1C	1C	1
2	1B	2
3	1C	2
4	1C	3
5	1C	4

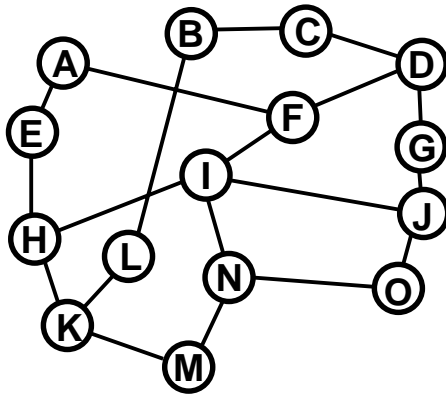
(c)

Fig. 5-17. Hierarchical routing.

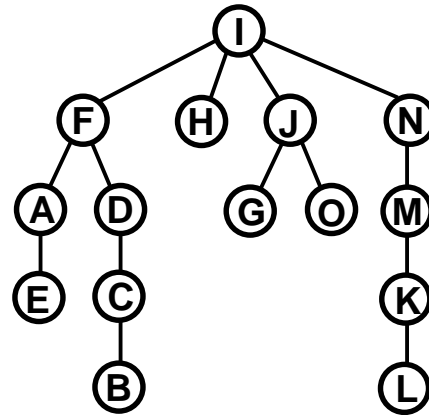
Routing broadcast: técnicas

- *Inundación*: fácil, pero se pueden producir bucles (limitar número de saltos al diámetro de la red).
- *Routing multidestino*: se manda un solo paquete con todas las direcciones de destino y en cada router se replica por las interfaces que corresponda.
- *Spanning tree*: si se conoce el árbol óptimo basta con seguirlo (posible con routing por el estado del enlace).
- *Encaminamiento por el camino inverso* (versión ‘barata’ del spanning tree): se examina la dirección de origen y la interfaz de entrada. Si esa interfaz es la ruta óptima para esa dirección se procede por inundación, si no se descarta (se considera entonces que es un duplicado). Se puede usar cuando no se conoce el spanning tree.

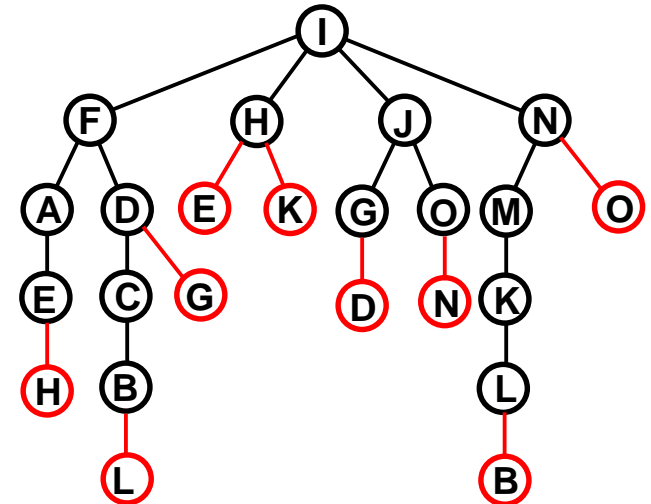
Distribución broadcast mediante el spanning tree y el encaminamiento por el camino inverso



Una red cualquiera



Encaminamiento broadcast mediante el spanning tree. Se distribuyen 14 paquetes

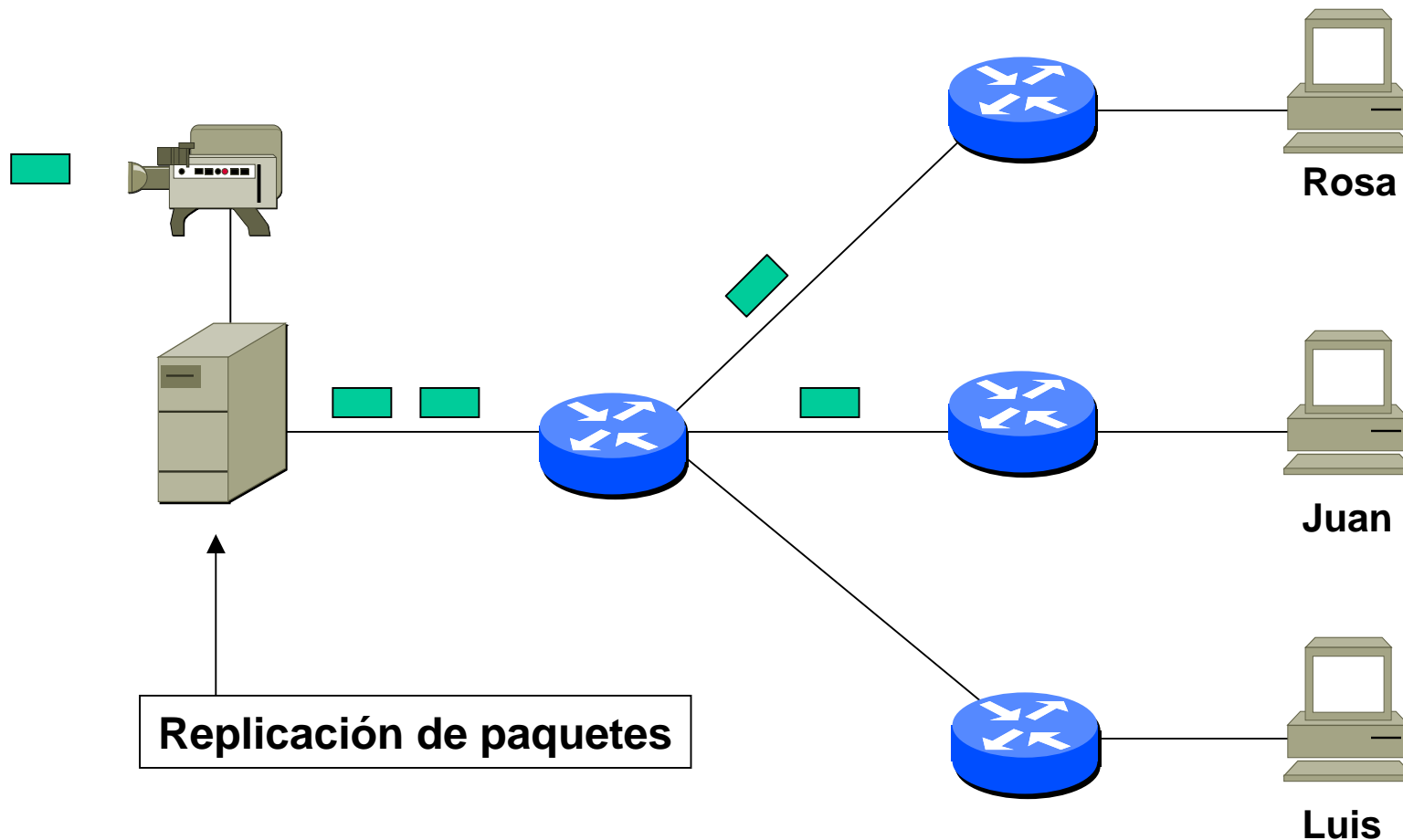


Uso del encaminamiento por el camino inverso para la distribución broadcast. Se distribuyen en total 23 paquetes, 9 duplicados (los duplicados aparecen en rojo)

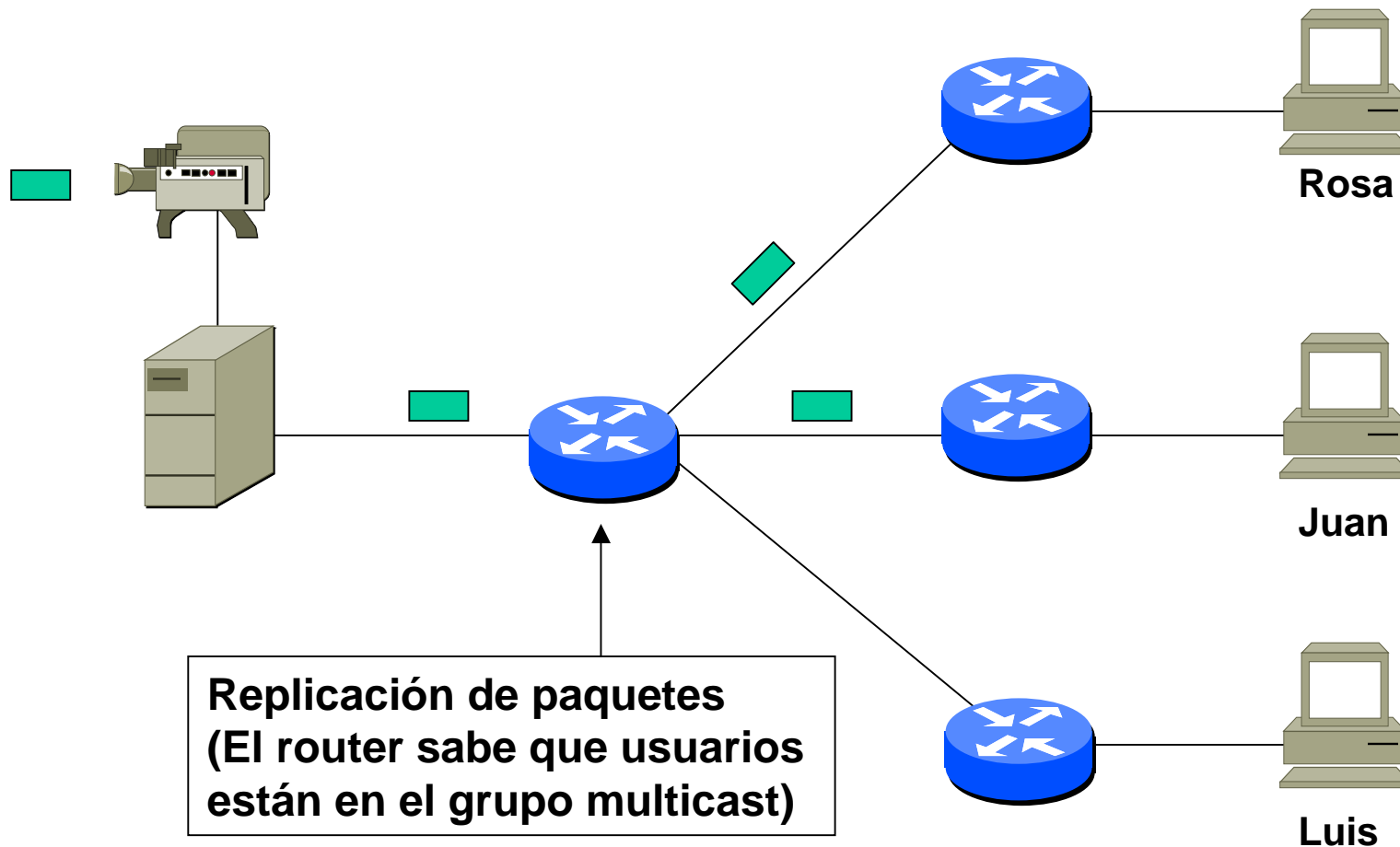
Routing multicast

- Permite hacer un uso eficiente de la red. Los paquetes solo llegan allí donde se necesitan y solo se envían una vez.
- Para emitir en un grupo multicast no es necesario pertenecer a él (solo para recibir).

Emisión Multicast en una red sin soporte Multicast



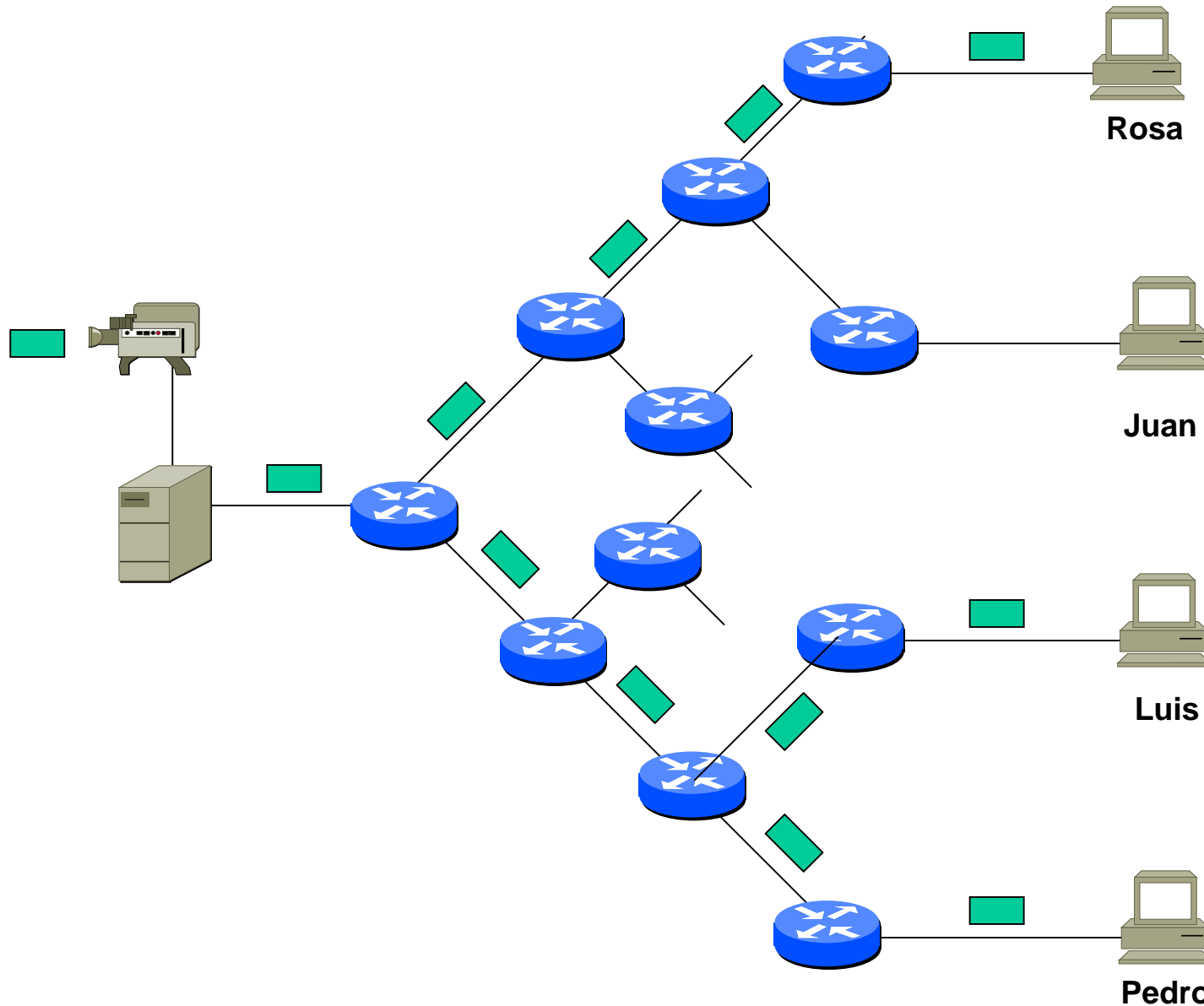
Emisión Multicast en una red con soporte Multicast



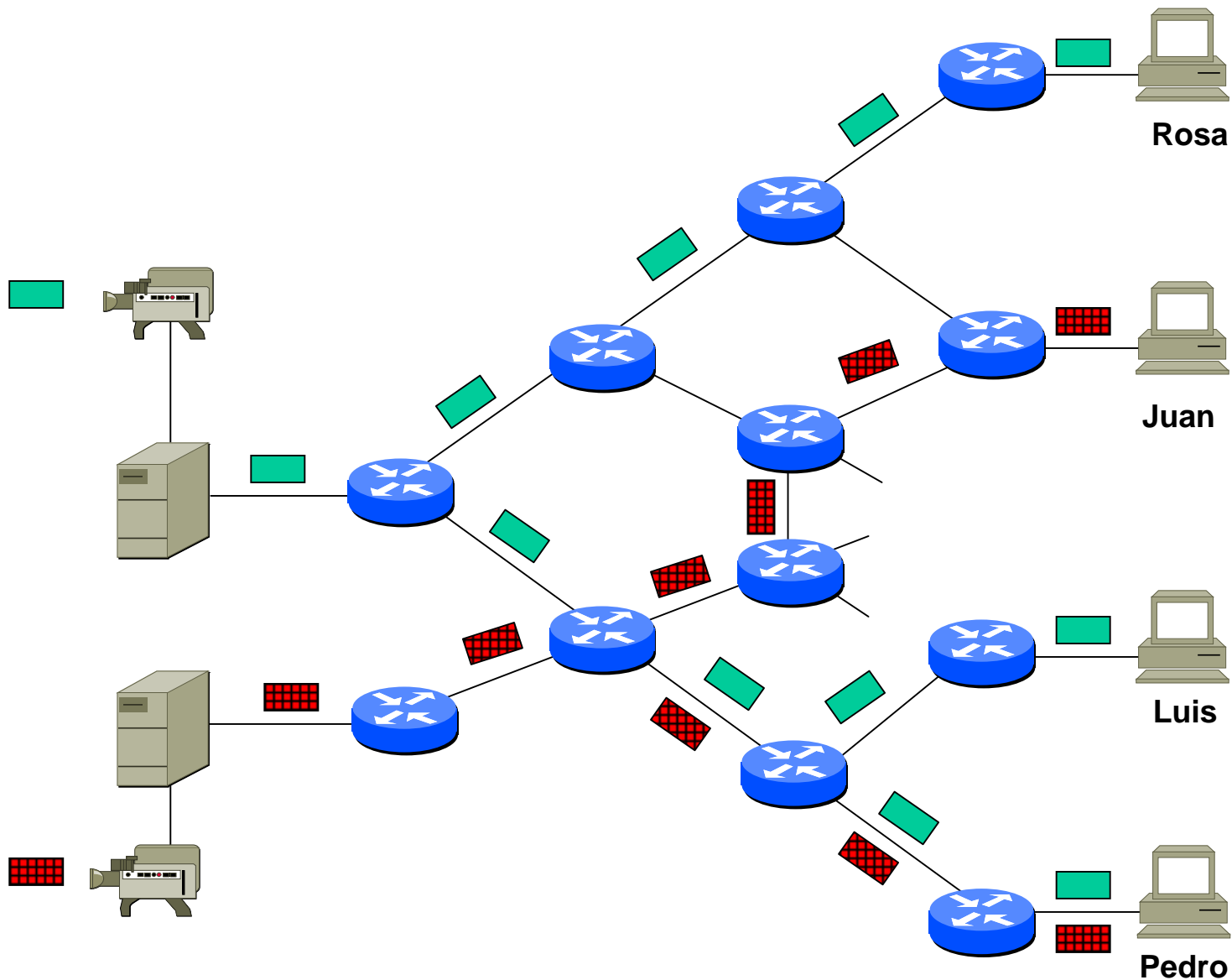
Routing multicast

- Cuando se usa routing por el estado del enlace el emisor construye el spanning tree con todos los miembros del grupo, colocándose él como raíz.
- En routing por vector distancia se suele usar encaminamiento por el camino inverso.
- La gestión del grupo multicast (altas o bajas) es independiente del routing. Los cambios en el grupo conllevan cambios en el árbol.

Emisión de un programa en una red multicast



Emisión de dos programas en una red multicast



Mbone: Multicast en Internet

- En Internet se han hecho pruebas de tráfico Multicast desde 1992, sobre todo para videoconferencias
- La red multicast se denomina MBone (Multicast Backbone)
- A partir de 1997 algunos ISP comerciales se han conectado a MBone

Problemas de MBone

- A pesar de sus ventajas MBone aun es todavía algo experimental. Razones:
 - Técnicamente el routing multicast es mucho más complejo que el routing unicast
 - Las aplicaciones AV que usan multicast no funcionan bien cuando hay congestión.
 - Las ventajas solo se presentan cuando el ISP tiene un número de usuarios elevado.
 - El enlace de la mayoría de los usuarios de ISPs no tiene capacidad para recibir las emisiones de MBone

Aplicaciones y Servicios en MBone

- Videoconferencia multipunto (utilizado regularmente desde 1992 para sesiones del IETF)
 - Vídeo (128 Kb/s – 3 Mb/s)
 - Audio (9-64 Kb/s)
 - Pizarra electrónica
- Servicios de vídeo ‘casi’ bajo demanda
 - Flujos MPEG-1, MPEG-2, MPEG-4, etc.
 - Aplicaciones de teleenseñanza (LANs empresas)

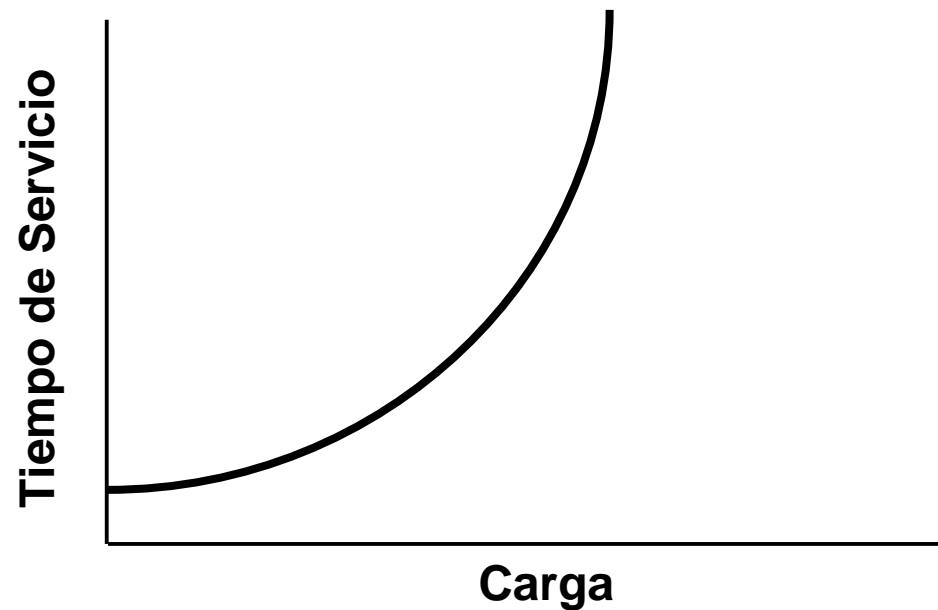
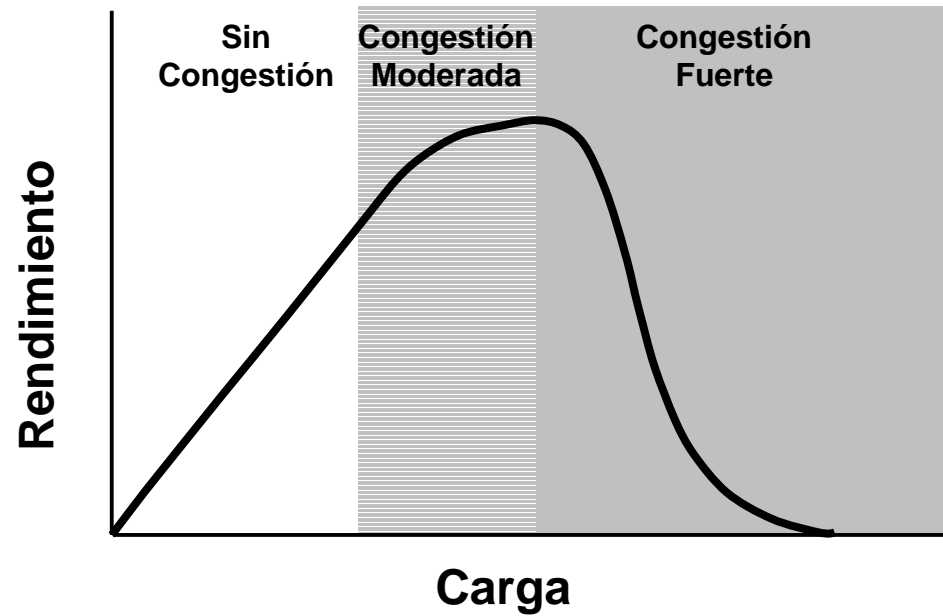
Sumario

- Aspectos generales del nivel de red
- Algoritmos de routing
- **Control de congestión**

Control de congestión

- Los tiempos de servicio aumentan de forma dramática cuando una línea o un router se aproxima a la saturación.
- No es posible ocupar una línea al 100% (tiempo de servicio infinito).
- Los buffers grandes permiten no descartar paquetes, pero aumentan el retardo. Esto puede causar retransmisiones y generar aún más tráfico.
- Cuando hay congestión severa el rendimiento global disminuye.

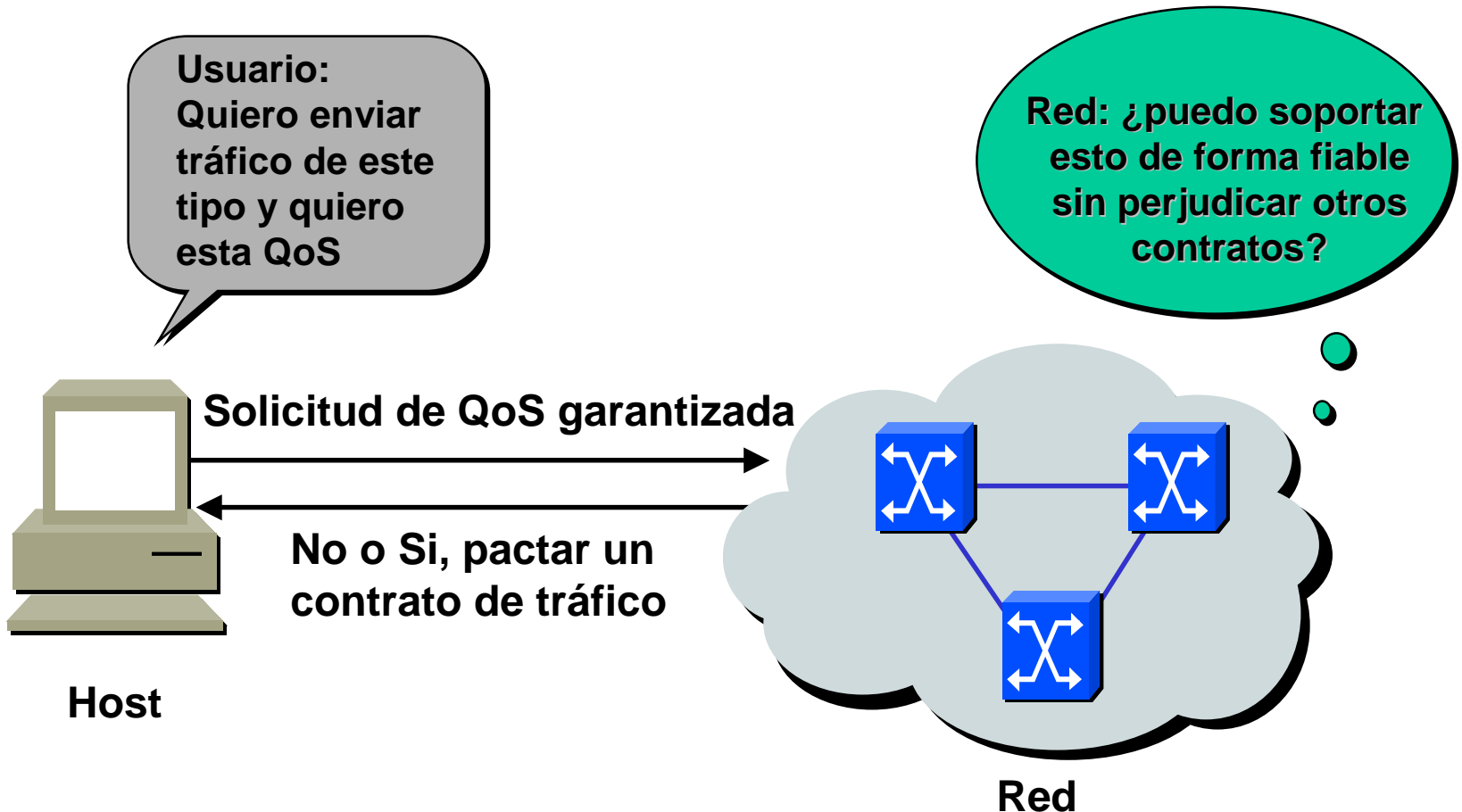
Efectos de la
congestión en el
rendimiento y el
tiempo de
servicio



Como evitar ‘de entrada’ la congestión

- *Cobrar* a los usuarios por el tráfico. Ej.: X.25
- Habilitar *recursos adicionales*. Ej.: RDSI
- Utilizar *rutas alternativas*. Ej.: routing dinámico.
- Imponer *límites* a los usuarios. Ej.: Frame Relay (CIR). Requiere labor de policía.
- ‘*Suavizar*’ las ráfagas. Ej. : ATM (pozal agujereado)
- *Planificar* caudales utilizados y reservar (QoS). Ej.: videoconferencia, aplicaciones multimedia.
- En redes CONS aplicar *control de admisión*. Ej.: ATM, red telefónica.

Control de Admisión



Si se supera el Control de Admisión la red y el usuario pactan un contrato de tráfico

Como detectar la congestión

- A nivel de red:
 - Porcentaje de paquetes descartados
 - Longitud media de las colas en las interfaces de los routers
- A nivel de transporte:
 - Retardo medio de los paquetes
 - Desviación media del retardo (jitter)
 - Porcentaje de paquetes perdidos (suponiendo que no se debe a errores)

Como detectar la congestión

- Normalmente los parámetros se monitorizan con fórmulas del tipo:

$$u_n = a u_{n-1} + (1-a) f$$

Donde:

f : valor instantáneo del parámetro medido

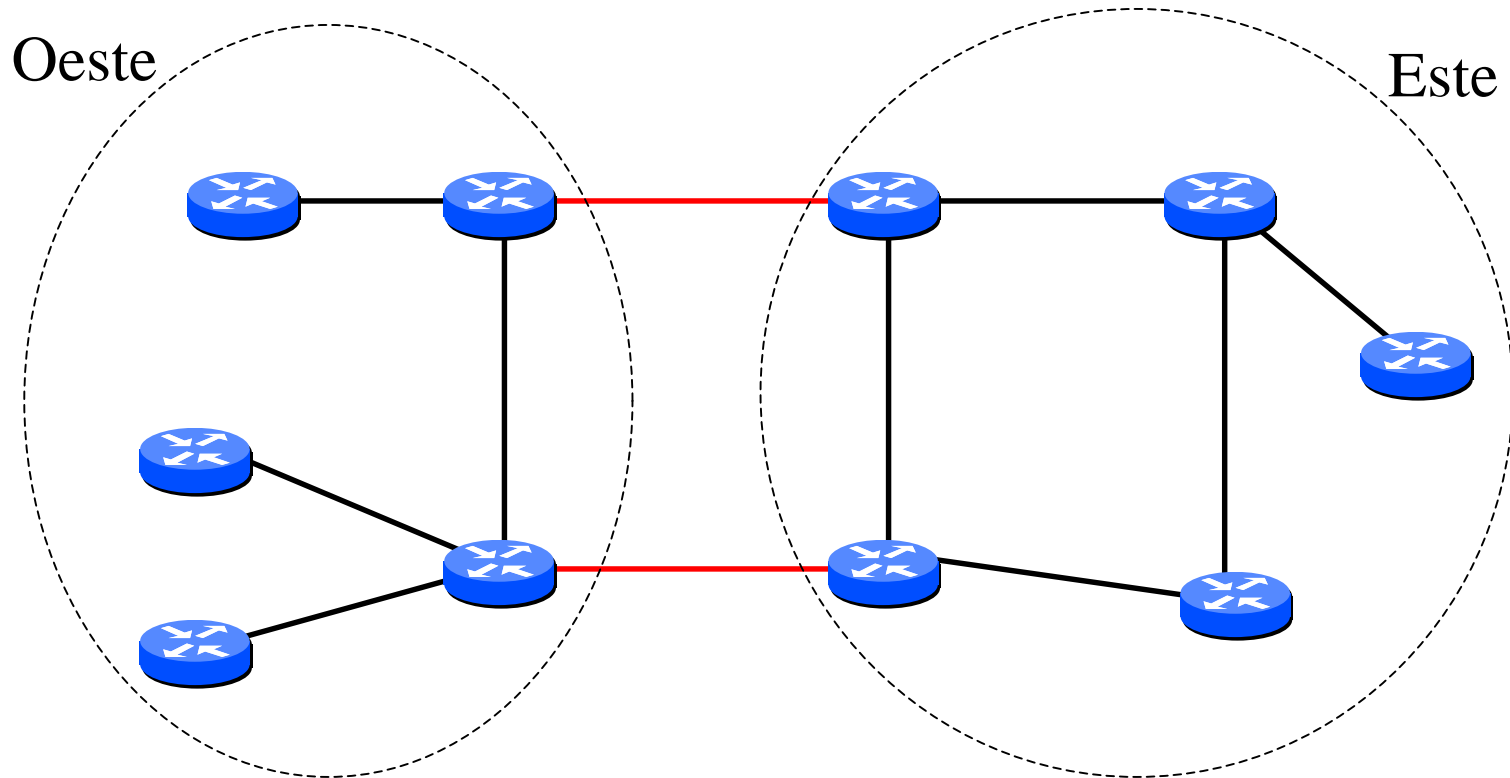
u_n : valor medio en la n-ésima iteración

a : constante para regular la inercia a los cambios

Así se consigue un efecto más amortiguado de los cambios y se evitan oscilaciones, Ej., con $a = 0,5$:

$$\text{Tr. medio}_n = 0,5 * \text{Tr. medio}_{n-1} + 0,5 * \text{Tr. instantáneo}$$

Redes ‘oscilantes’



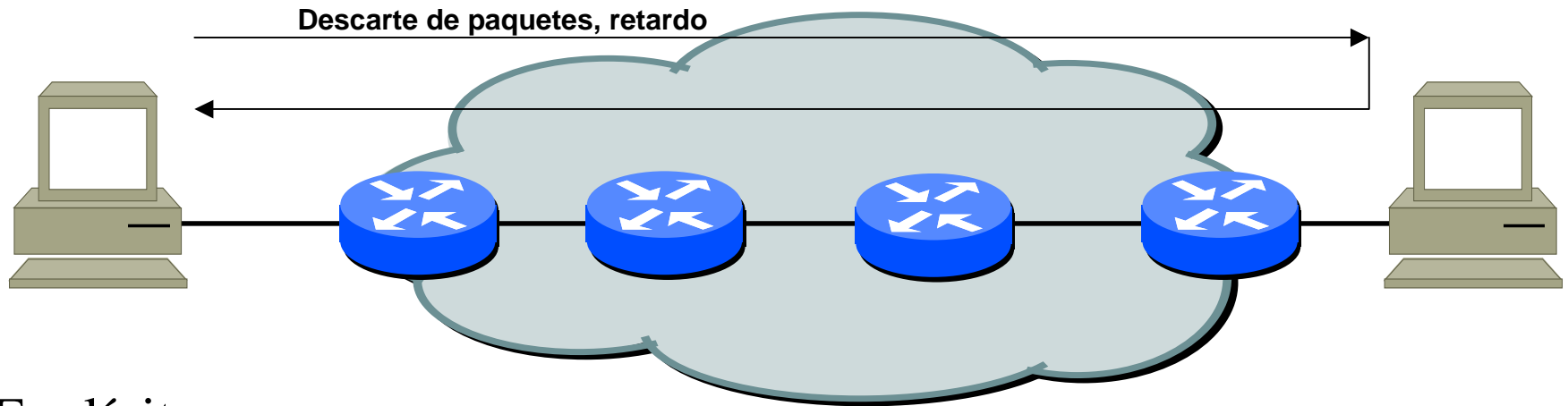
En esta topología el tráfico entre las dos redes se reparte por los dos enlaces marcados en rojo. Esto puede producir situaciones oscilantes al elegir como ruta óptima uno u otro camino para todo el tráfico

Como notificar una situación de congestión

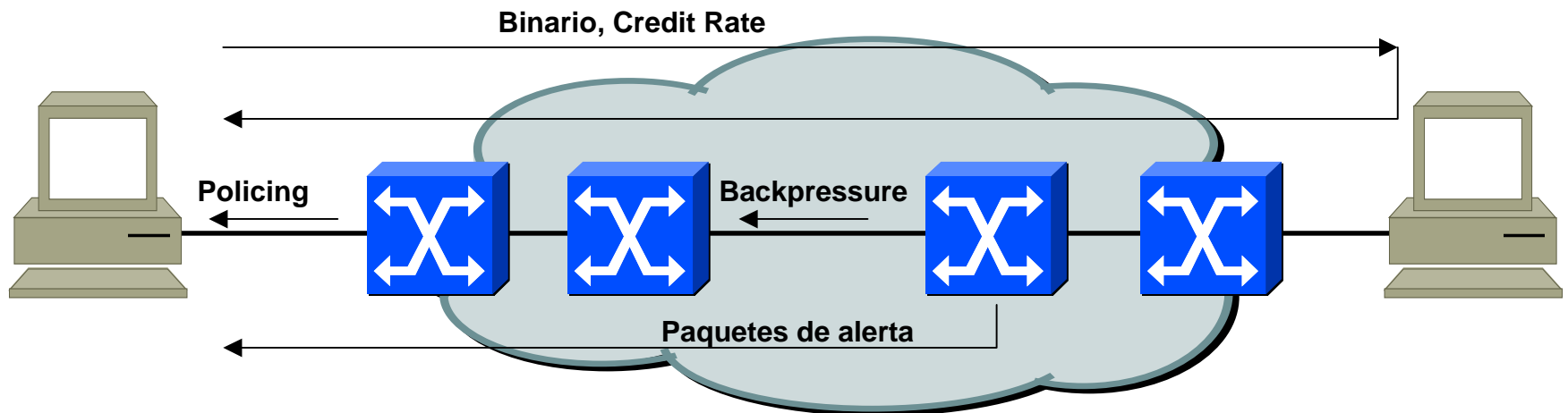
- Notificación implícita (descarte de paquetes):
 - El emisor bajará el ritmo (supondrá que se han perdido por congestión). Ej. TCP/IP.
- Notificación explícita (avisos al emisor):
 - Paquetes informativos o de alerta enviados al emisor por los routers. Ej. ATM, Frame Relay.
 - Aviso ‘piggybacked’ en un paquete de datos dirigido al emisor. (ATM, Frame Relay).
 - Aviso ‘piggybacked’ en un paquete de datos dirigido al receptor para que avise al emisor. (Frame Relay).

Mecanismos de control de congestión

Implícitos:



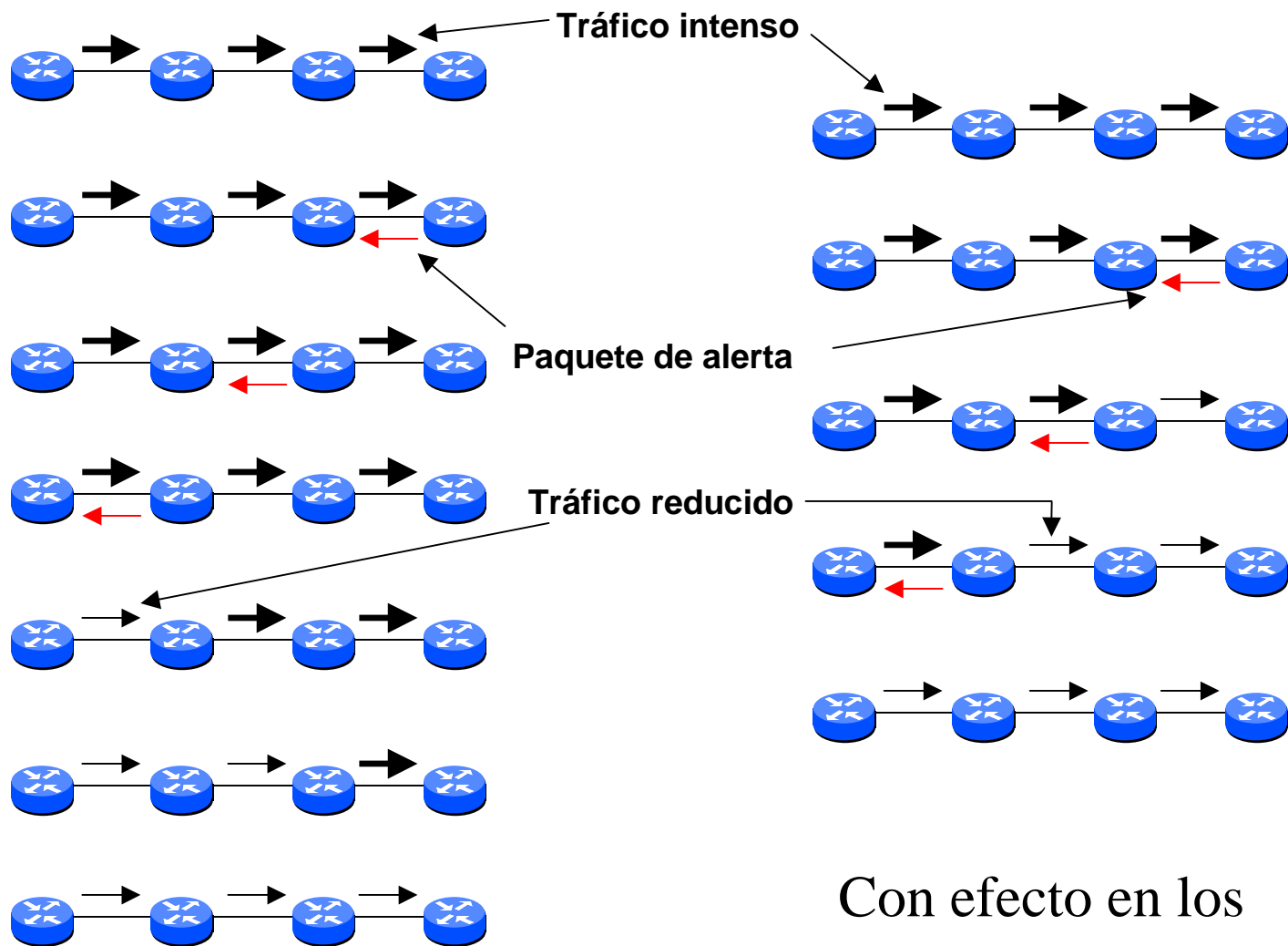
Explícitos:



Medidas ante una situación de congestión

- Reducir o congelar el envío de paquetes de los hosts hasta que no haya congestión.
- En algún caso los routers intermedios pueden ayudar reteniendo parte de los paquetes en sus buffers.
- Descartar paquetes. A veces estos llevan alguna indicación de su importancia para el descarte (paquetes de 1ª y 2ª clase).
- Descarte inteligente, ej.: si se descarta un fragmento descartar también los demás.

Paquetes de alerta ('choque')



Con efecto en los
nodos intermedios

Con efecto en el emisor

Problemas del control de congestión

- Se pueden dar situaciones cíclicas que impidan un aprovechamiento eficiente de los recursos (todos los hosts bajan el ritmo a la vez).
- Algunos creen que la lucha contra la congestión por mecanismos sofisticados (explícitos) es una batalla perdida (los remedios llegarán demasiado tarde para ser útiles).
- El uso de notificación explícita puede agravar aún más el problema.

Perfil de tráfico y vigilancia

- *Perfil de tráfico o conformado de tráfico (traffic shaping)*: condiciones máximas de uso de la red que el usuario se compromete a cumplir con el proveedor del servicio.
- *Vigilancia de tráfico (traffic policing)*: labor de monitorización que el proveedor realiza para asegurarse que el usuario cumple su palabra.
- Si el usuario incumple el proveedor puede:
 - a) Descartar el tráfico no conforme
 - b) Marcarlo como de ‘segunda clase’ y pasarlo a la red, o
 - c) Pasarlo a la red sin mas (no es habitual)

Pozal agujereado (leaky bucket)

- El pozal agujereado se utiliza para suavizar las ráfagas (traffic shaping) y para asegurar que el tráfico introducido es el acordado (traffic policing).
- El usuario dispone de un caudal constante ρ en la red (el agujero) y un buffer de una capacidad C (el pozal) que absorberá las ráfagas que produzca.
- Si el buffer se llena el tráfico excedente se considera no conforme. Normalmente se descarta o se pasa como tráfico de ‘segunda’ clase.

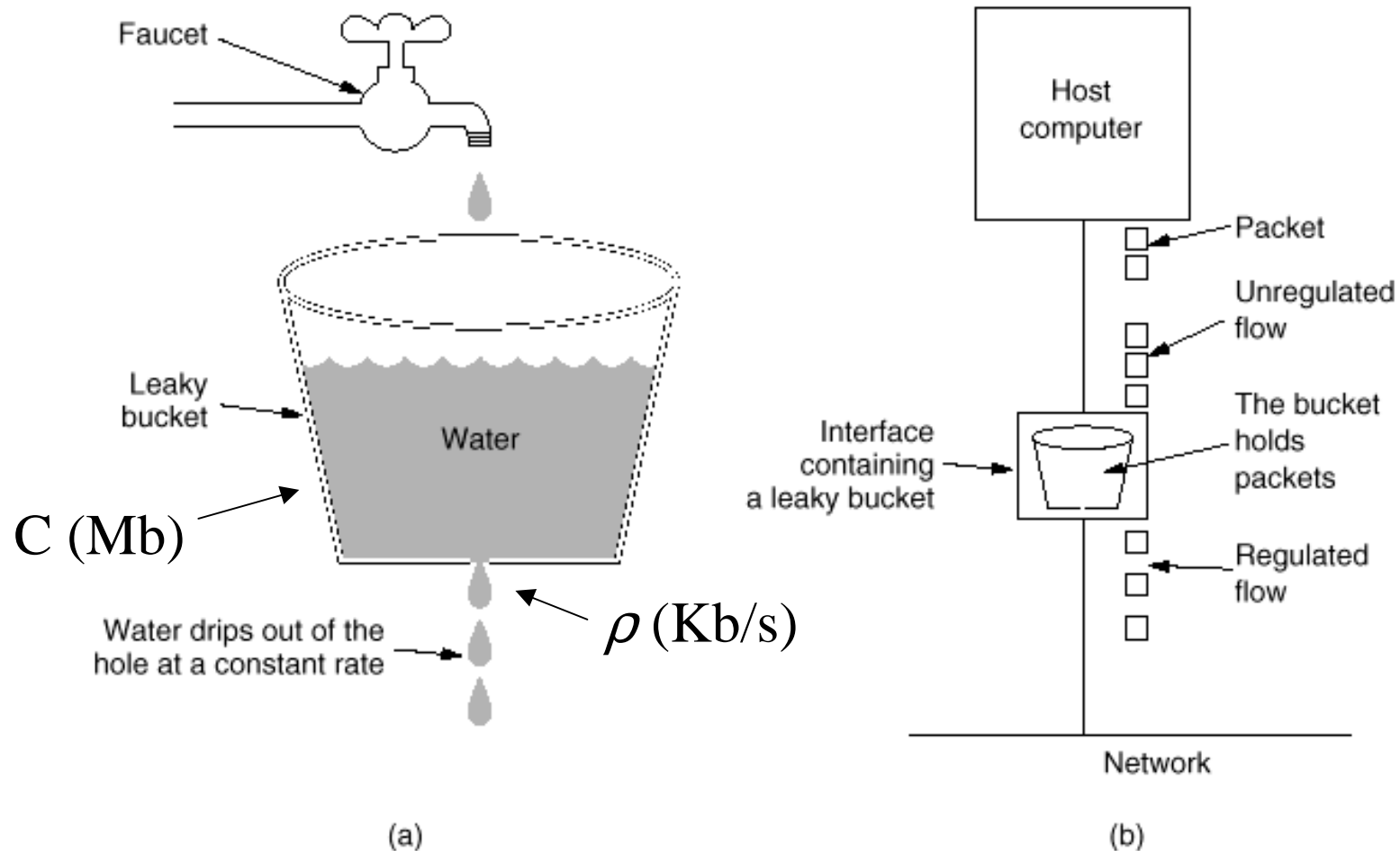


Fig. 5-24. (a) A leaky bucket with water. (b) A leaky bucket with packets.

Ejemplo de funcionamiento de un pozal agujereado

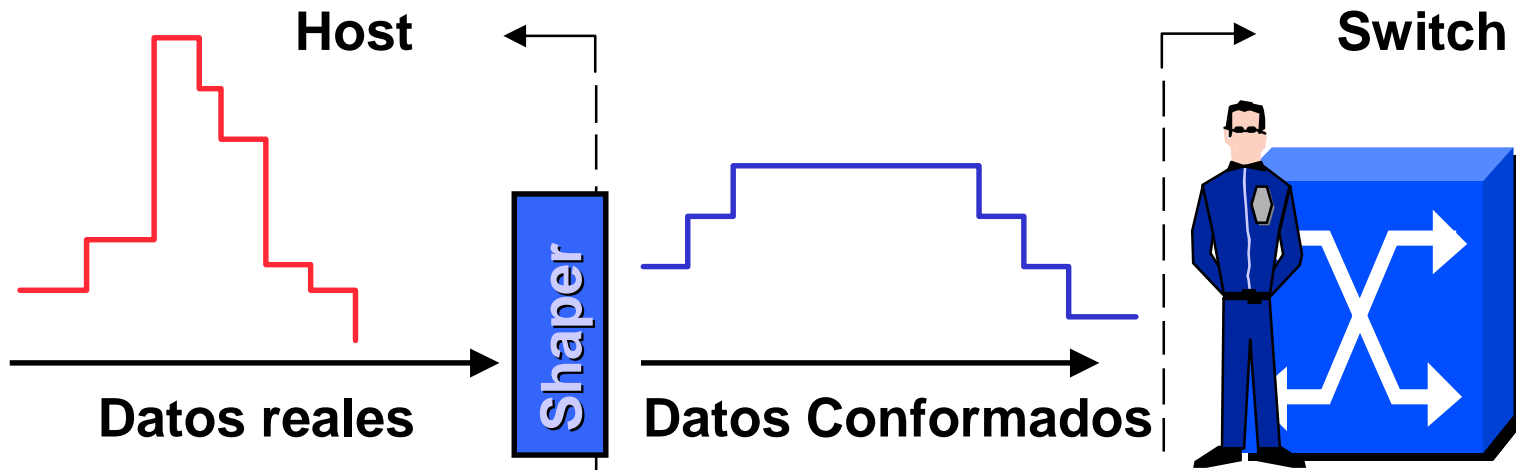
Parámetros: $\rho = 20 \text{ Mb/s}$, $C = 10 \text{ Mbits}$

Ráfaga de 10 Mbits recibida en 50 ms

Instante	Tr. Entrado	Tr. Salido	En pozal
0 ms	0	0	0
10 ms	2 Mb	0,2 Mb	1,8 Mb
20 ms	4 Mb	0,4 Mb	3,6 Mb
30 ms	6 Mb	0,6 Mb	5,4 Mb
40 ms	8 Mb	0,8 Mb	7,2 Mb
50 ms	10 Mb	1,0 Mb	9 Mb
60 ms	10 Mb	1,2 Mb	8,8 Mb
70 ms	10 Mb	1,4 Mb	8,6 Mb
80 ms	10 Mb	1,6 Mb	8,4 Mb
...			
500 ms	10 Mb	10 Mb	0 Mb

← Máximo

Traffic Shaping y Traffic Policing



**Conformado de Tráfico:
Cumplir el contrato**

- Algoritmo del pozo agujereado
- Limitar pico y tamaño de ráfagas
- Limitar retardo y jitter

**Vigilancia de Tráfico: Vigilar
y obligar su cumplimiento**

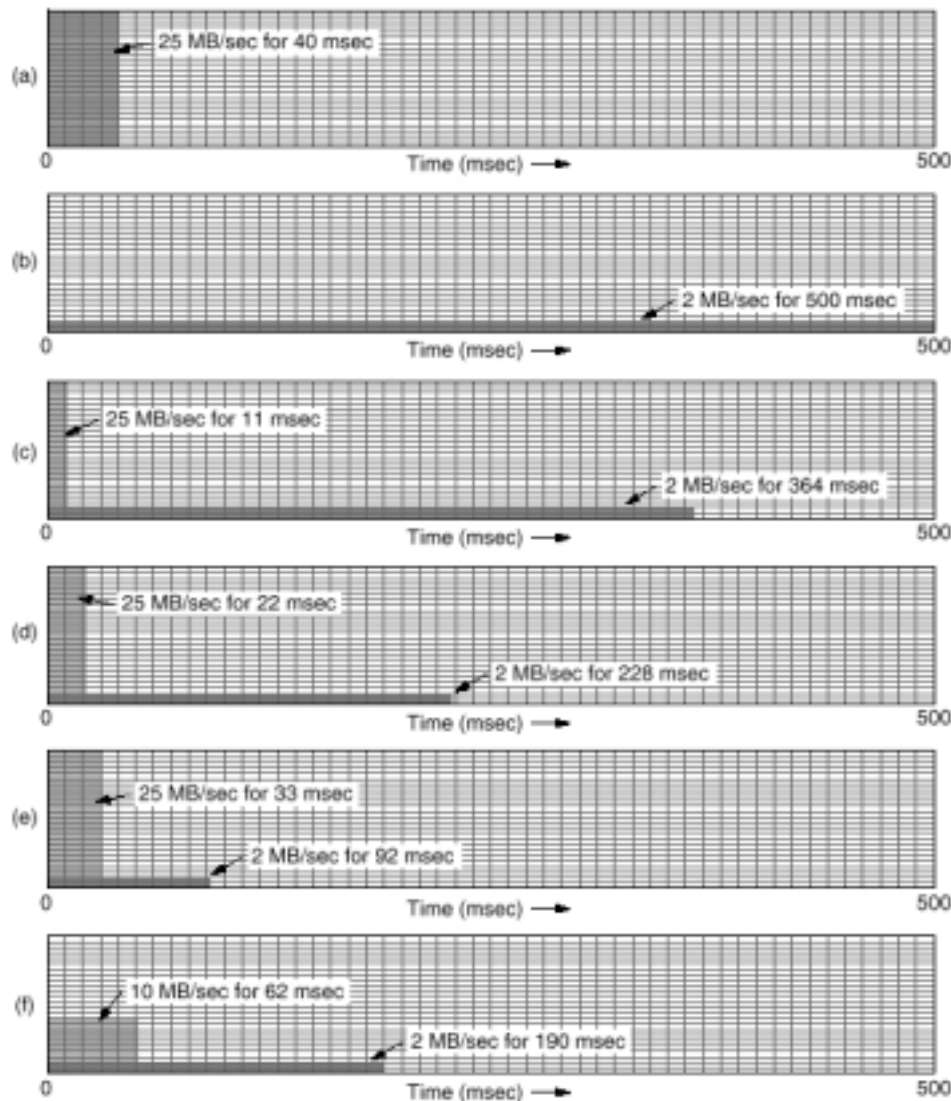
¿El tráfico recibido cumple el contrato?

Si no:

- Puede marcar celdas de 2ª clase
- Puede descartar celdas de 2ª clase

Pozal con crédito (token bucket)

- El pozal agujereado no fomenta el ‘ahorro’. Un usuario que no ha transmitido nada recibe el mismo servicio que uno que siempre ha transmitido con el caudal ρ .
- El pozal con crédito soluciona esto. Funciona así:
 - Cuando el usuario no transmite el pozal suma créditos.
 - El máximo crédito acumulable es igual a la capacidad del pozal.
 - Los créditos permiten transmitir con un caudal mayor (M) hasta que se consumen (normalmente M es la velocidad de la interfaz).
 - Cuando no hay crédito funciona como un pozal agujereado.
- A menudo se combina un pozal con crédito seguido de un pozal agujereado de caudal mayor



Ráfaga

Pozal agujereado

Pozal con crédito (25%)

Pozal con crédito (50%)

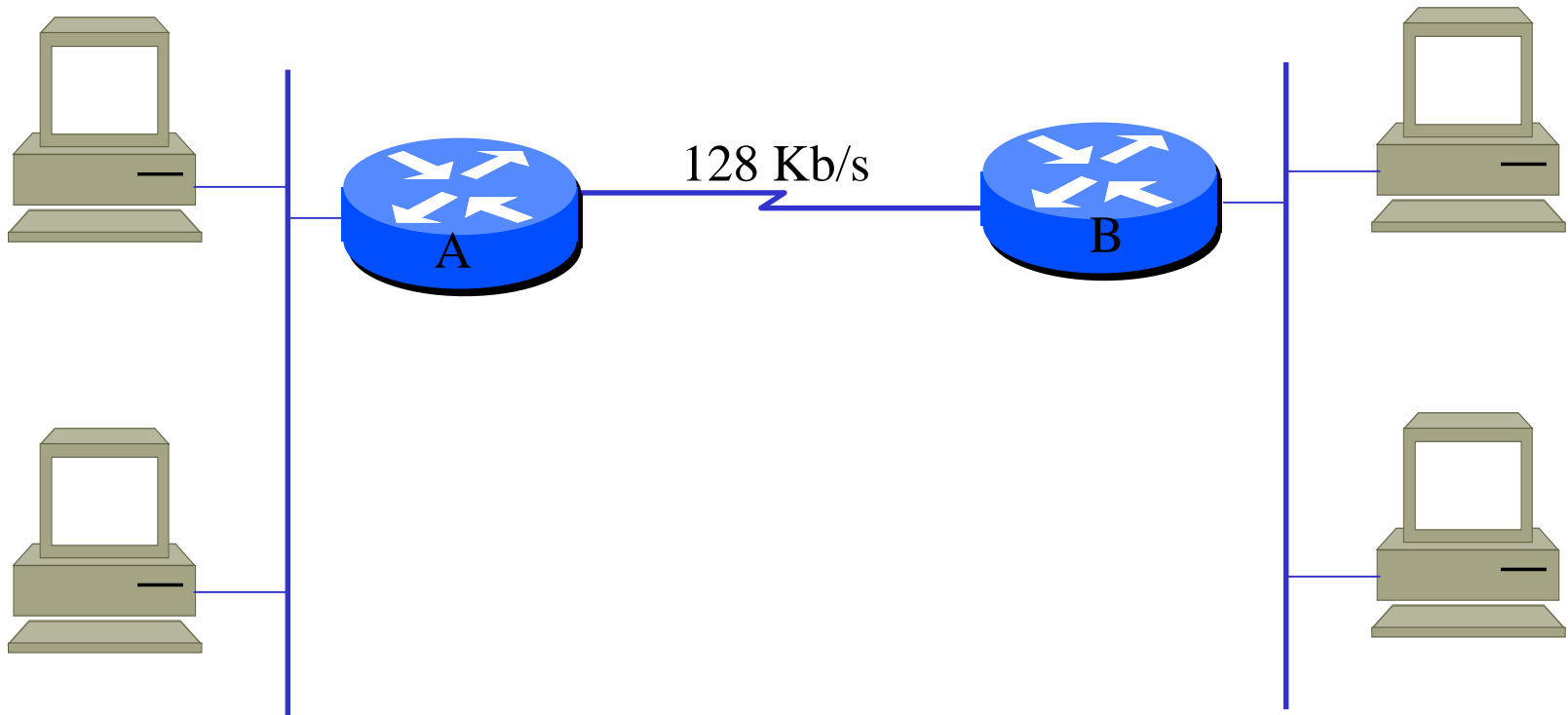
Pozal con crédito (75%)

Pozal con crédito seguido
de pozal agujereado

Fig. 5-25. (a) Input to a leaky bucket. (b) Output from a leaky bucket. (c) - (e) Output from a token bucket with capacities of 250KB, 500KB, and 750KB. (f) Output from a 500KB token bucket feeding a 10 MB/sec leaky bucket.

Ejercicios

Ejercicio 4



Paquetes de 164 bytes (1312 bits)

Audioconferencia: 1 paquete cada 40 ms, 25 paquetes/s

Calcular caudal máximo para que el retardo no supere 80 ms

Ejercicio 4

Aproximamos retardo a Tiempo de servicio, T.

Por teoría de colas: $T = 1 / (\mu c - \lambda)$, donde:

$\mu = 1 / \text{tamaño-paquete}$

c : velocidad del enlace (bits/s)

μc : capacidad del enlace (paquetes/s)

λ : Caudal medio (paquetes/s)

En este caso:

$$\mu = 1 / 1312 = 0,000762 \text{ bits}^{-1}$$

$$c = 128.000 \text{ bits/s}$$

$$\mu c = 97,56 \text{ pps}$$

$$T \leq 0,08 \text{ seg}$$

Ejercicio 4

El caudal máximo tolerable será el que dé $T = 0,08$ seg.
Cuando hay una audioconferencia en marcha:

$$0,08 = 1 / (97,56 - x - 25)$$

25: paquetes producidos por la audioconferencia

x: paquetes producidos por otras aplicaciones

Despejando $x = 60,06$ pps $\rightarrow 78,8$ Kb/s $\rightarrow 61,6\%$ ocupac.
(49 Kb/s libres)

Para línea de 2048 Kb/s:

$$0,08 = 1 / (1560,98 - x - 25)$$

$x = 1523$ pps $\rightarrow 1998,8$ Kb/s $\rightarrow 97,6\%$ ocupac.
(49 Kb/s libres)

Ejercicio 4

