



**SERVICIOS EN LA WEB Y DISTRIBUCIÓN DE CONTENIDOS**  
*Área de Ingeniería Telemática*

# *Streaming*

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

Programa de Tecnologías para la gestión distribuida de la  
información



# Contenido

- Introducción: ¿Streaming?
- Arquitectura del servicio
- Necesidades:
  - En el servidor
  - En la red



SERVICIOS EN LA WEB Y DISTRIBUCIÓN DE CONTENIDOS  
Área de Ingeniería Telemática

# Multimedia

## ¿A qué llamamos *Streaming*?



# Multimedia Networking Applications

## Características principales:

- Sensibles al retardo
  - end-to-end delay
  - delay jitter
- Toleran pérdidas: si son infrecuentes solo causan pequeños *glitches*
- Requisitos diferentes de la transferencia de datos (ficheros)

***Jitter*** es la variabilidad de los retardos dentro del mismo flujo de paquetes

## Clase de aplicaciones multimedia

- “Streaming” de audio o vídeo grabado
- Streaming de audio o vídeo en vivo
- Audio o vídeo interactivo

## Compresión (Audio)

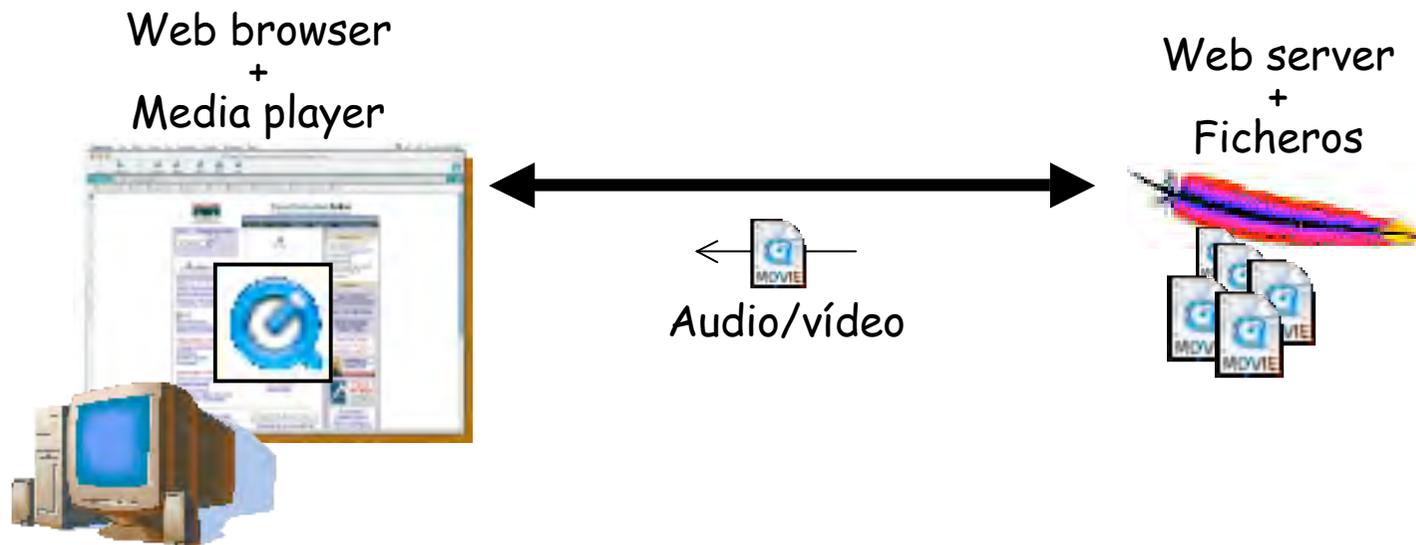
- CD: 1.411 Mbps
- MP3: 96, 128, 160 kbps
- Internet telephony: 5.3-13 kbps

## Compresión (Vídeo):

- MPEG 1 (CD-ROM) 1.5 Mbps
- MPEG2 (DVD) 3-6 Mbps
- MPEG4 (< 1 Mbps)



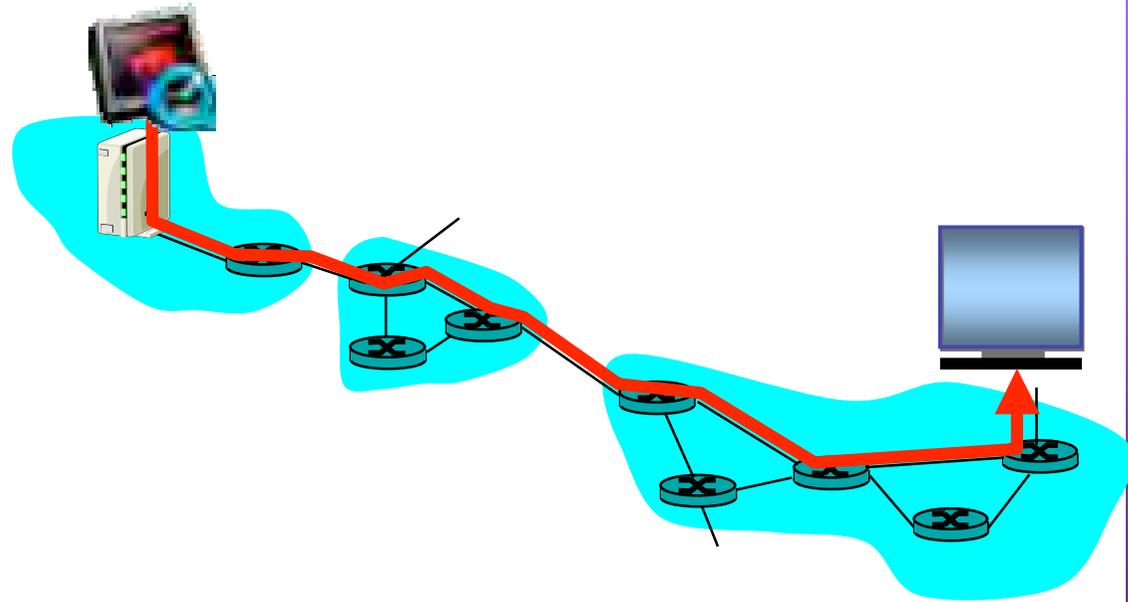
# Download+play vs “Streaming”



- Ficheros de audio o vídeo almacenados en servidor
- Ficheros transferidos como objetos HTTP
  1. Download+play
    - Recibidos completamente por el cliente
    - Pasados al reproductor
    - No hay “streaming”. Gran espera hasta empezar
  2. “Streaming”
    - Pasados al reproductor a medida que los recibe
    - Calcula cuándo empezar dada la velocidad a la que recibe
    - “Streaming” pero recibe a la máxima velocidad (transferencia HTTP)



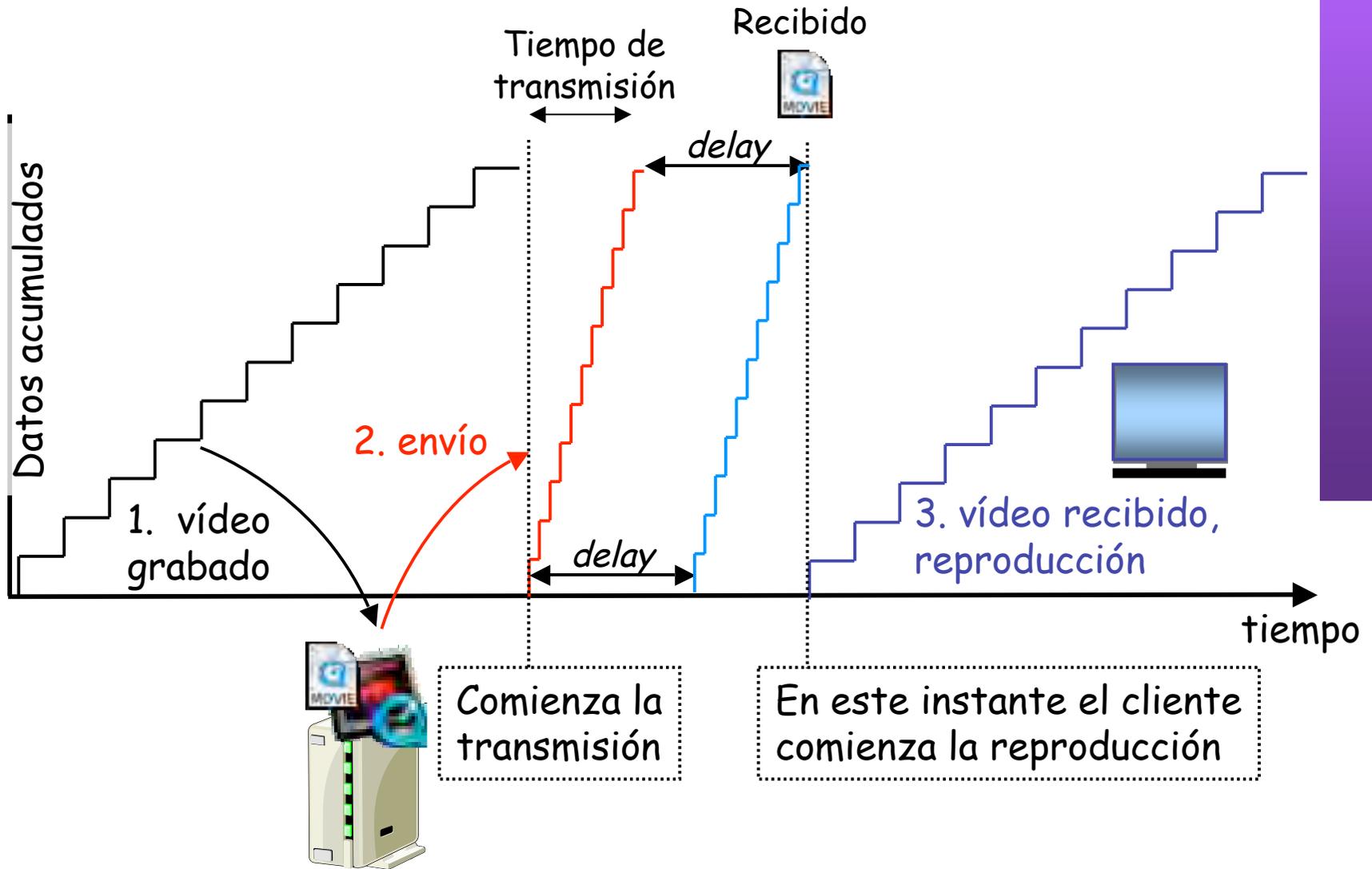
# “Streaming” de multimedia en disco



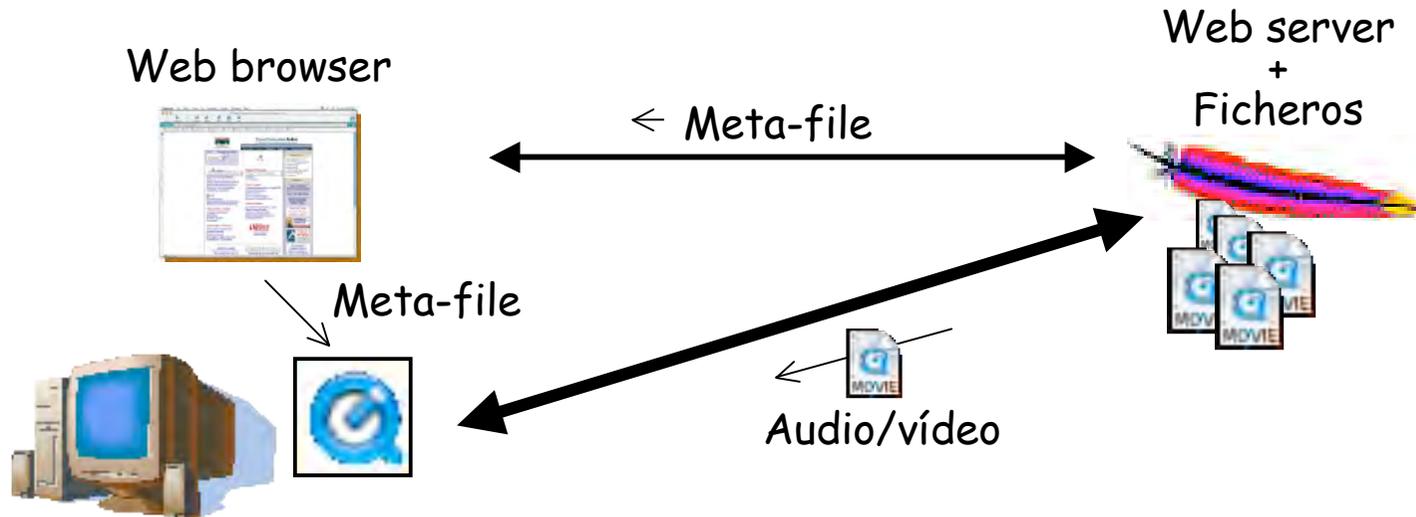
- “Streaming”
  - El cliente comienza la reproducción antes de recibir todos los datos
  - Es más bien un “play-as-you-get”
  - Requisitos temporales para los datos aún por recibir
- Streaming
  - La velocidad de transmisión es aproximadamente la de consumir los datos



# Download+play



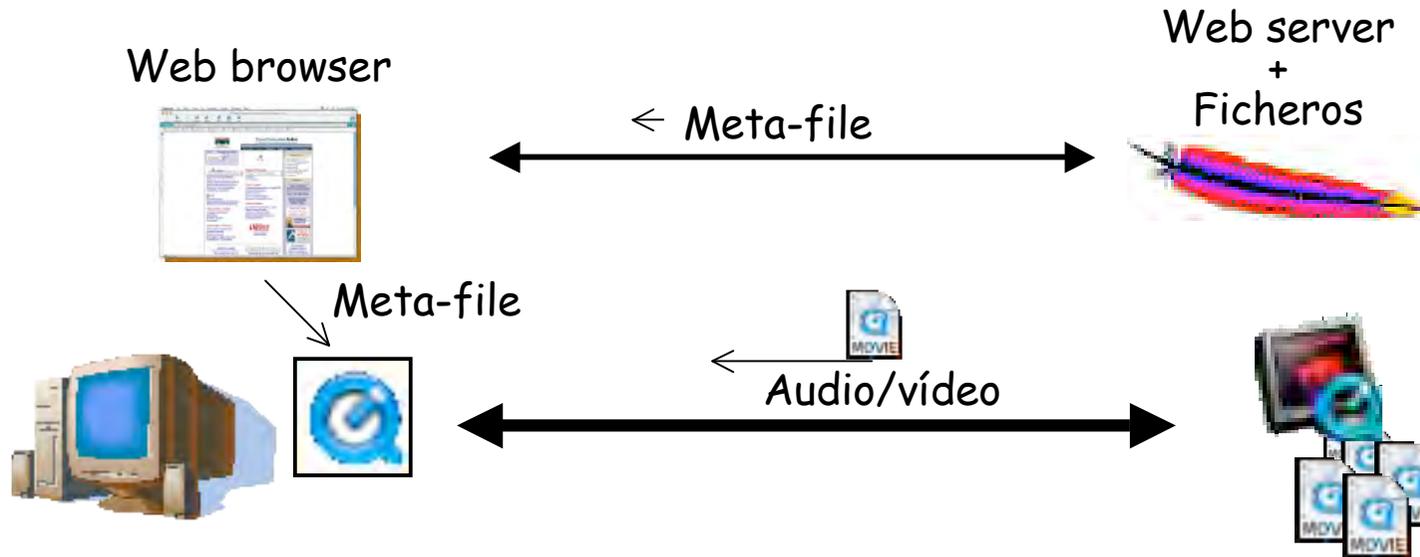
# Play-as-you-get



- Ficheros de audio o vídeo
- El navegador obtiene un *meta-file* que describe cómo conseguir la película
- Le pasa ese *meta-file* al reproductor
- El reproductor contacta con el servidor y solicita el fichero
- El servidor se lo envía (HTTP)



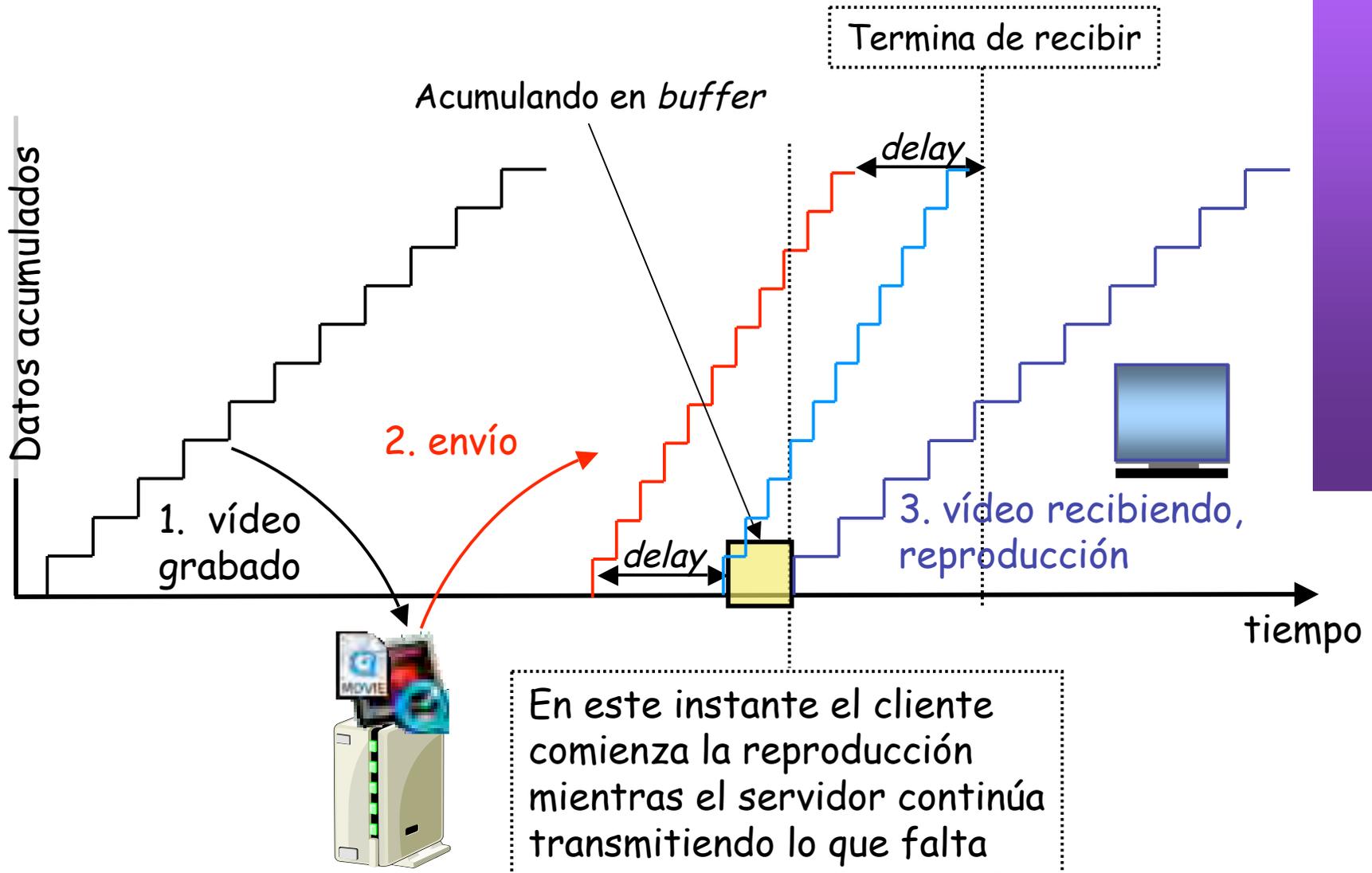
# Play-as-you-get



- Permite que se emplee un protocolo diferente de HTTP
- Permite emplear UDP en vez de TCP

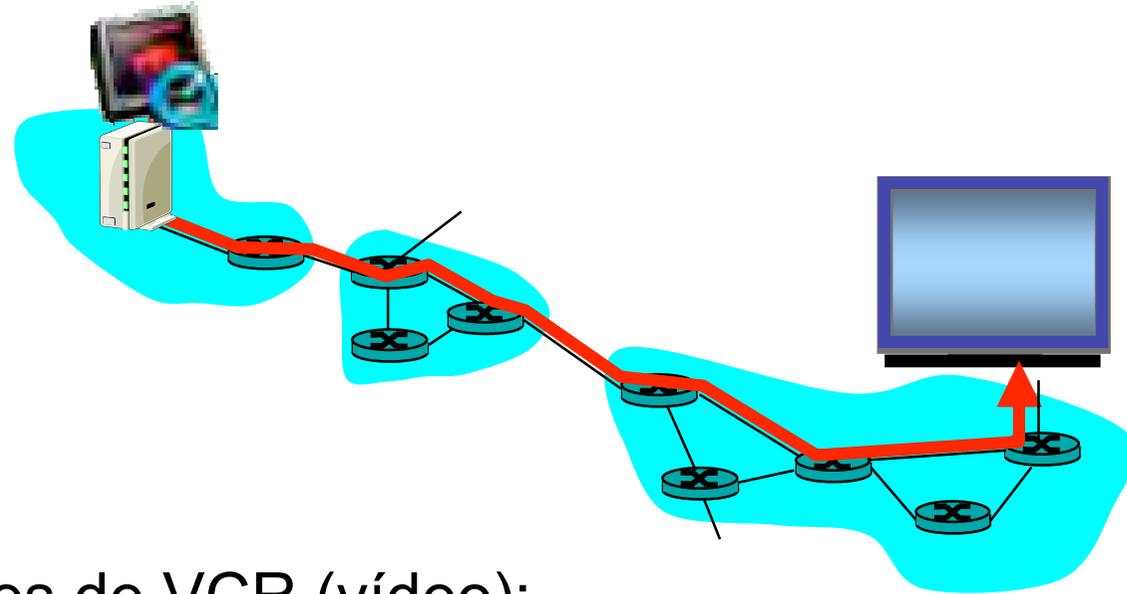


# Play-as-you-get





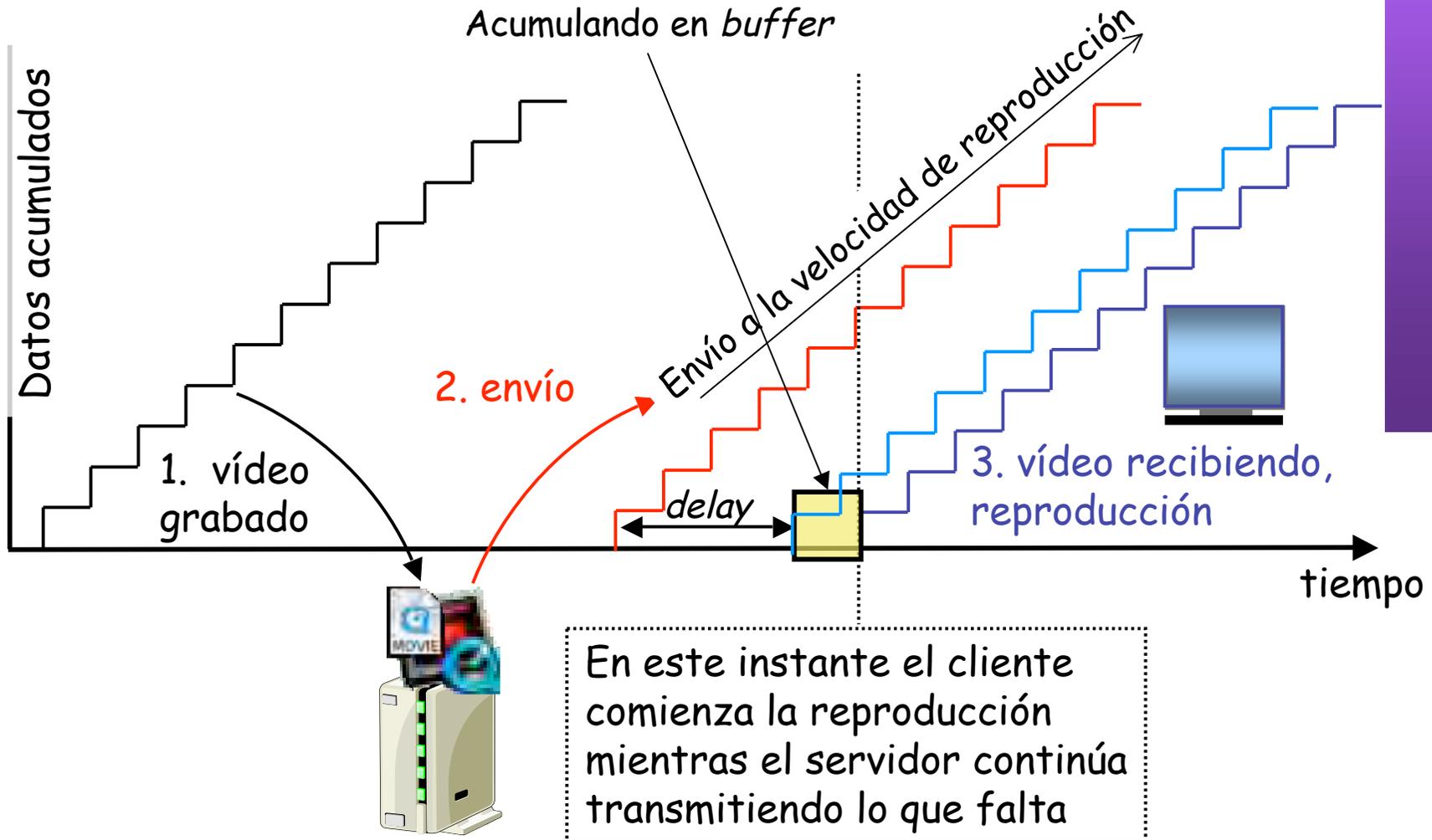
# “Streaming” de multimedia en disco: Interactividad



- Funcionalidades de VCR (vídeo):
  - Pausa, rebobinar, avance rápido, etc.
  - Retardo de comienzo 10 segs OK
  - Retardo ante un comando 1-2 seg OK
  - Protocolos para estos comandos (RTSP)

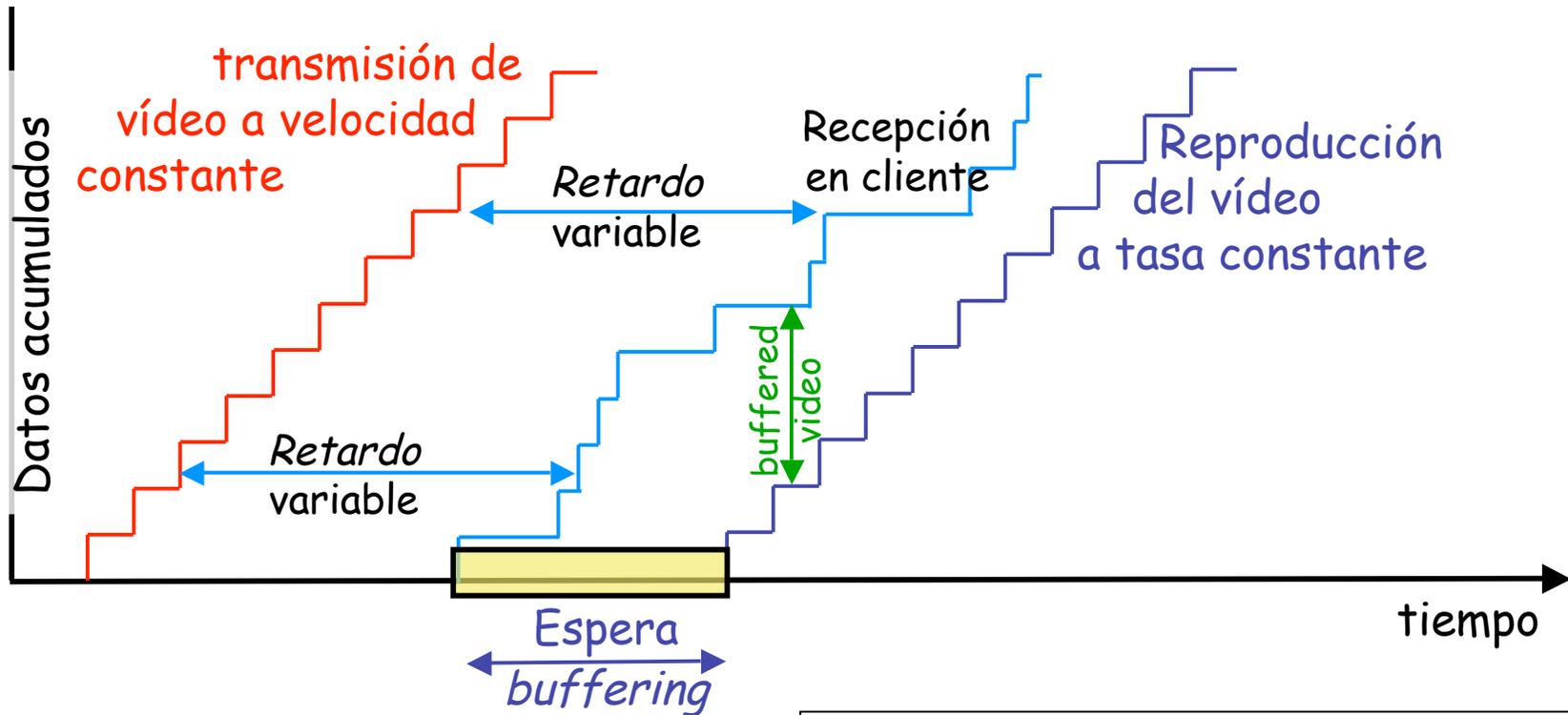


# Streaming

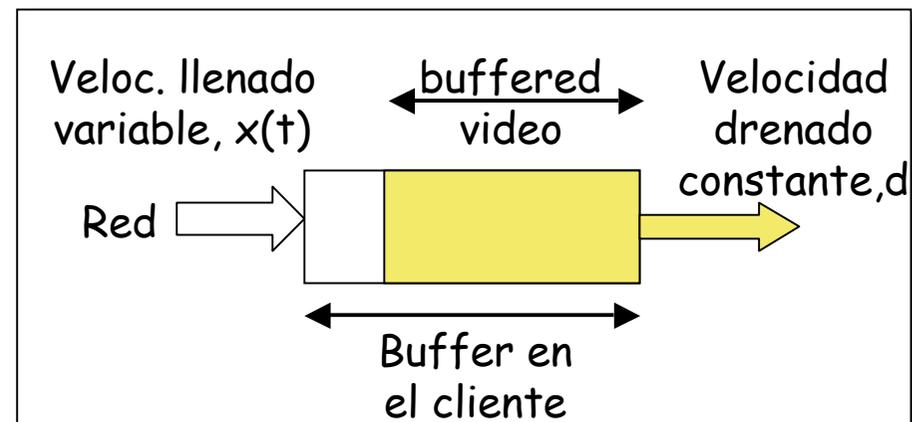




# Streaming: Client Buffering



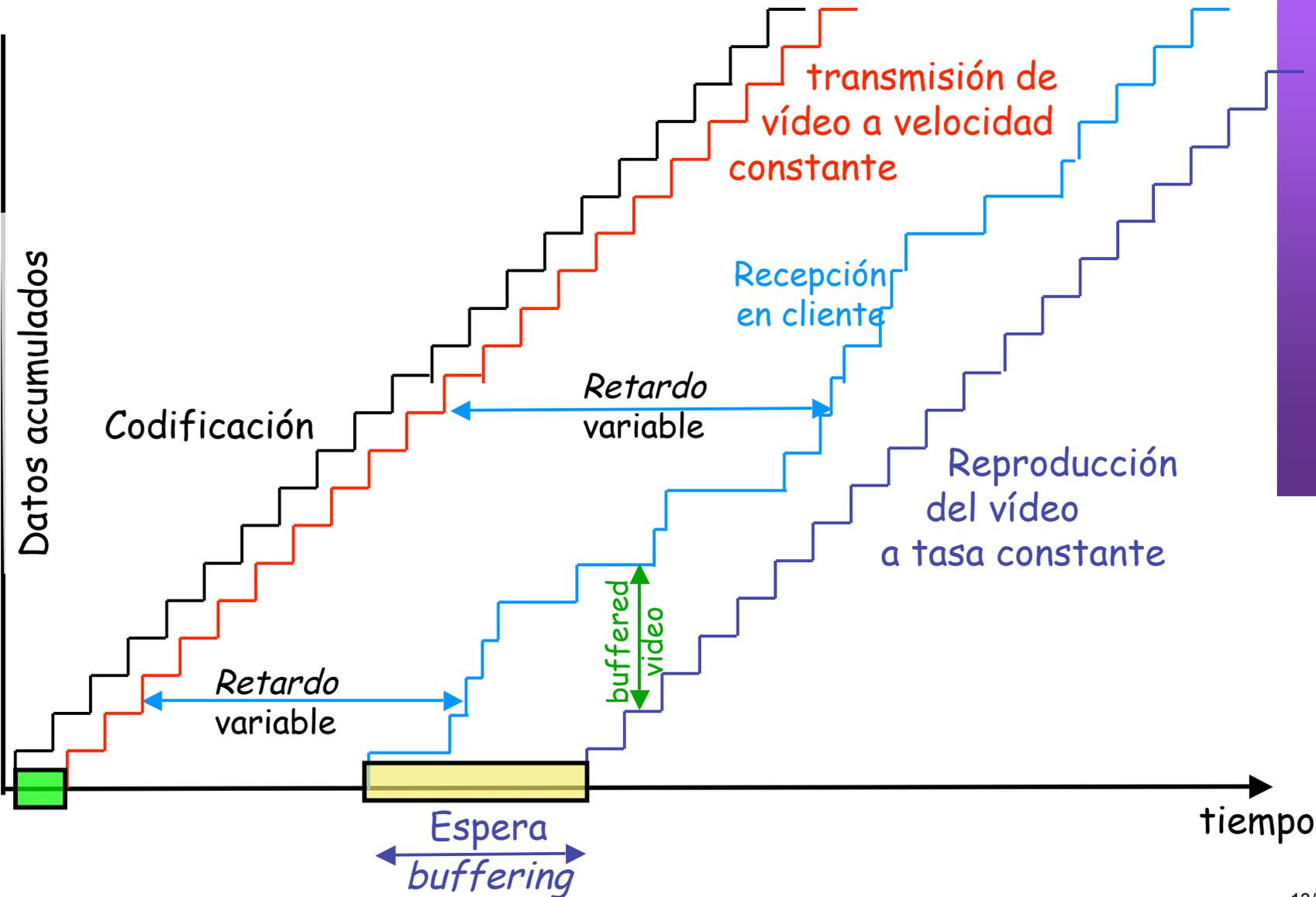
- El cliente acumula parte del vídeo en buffer antes de comenzar
- Permite absorber variaciones en el retardo en la red (*jitter*)
- Si el buffer se vacía el reproductor se detiene





# Live Streaming

- Avance rápido imposible





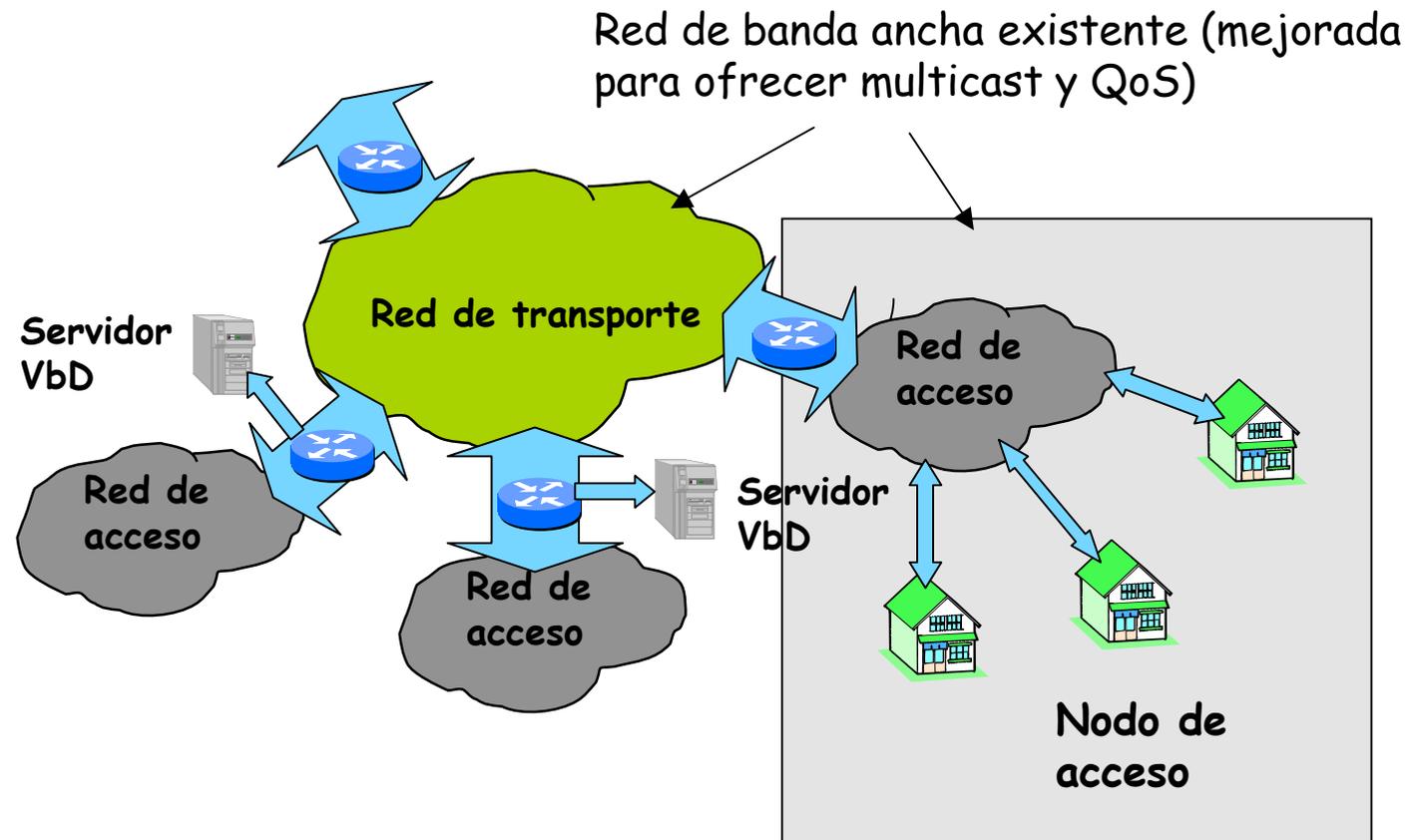
**SERVICIOS EN LA WEB Y DISTRIBUCIÓN DE CONTENIDOS**  
*Área de Ingeniería Telemática*

# Arquitecturas



# Arquitectura general

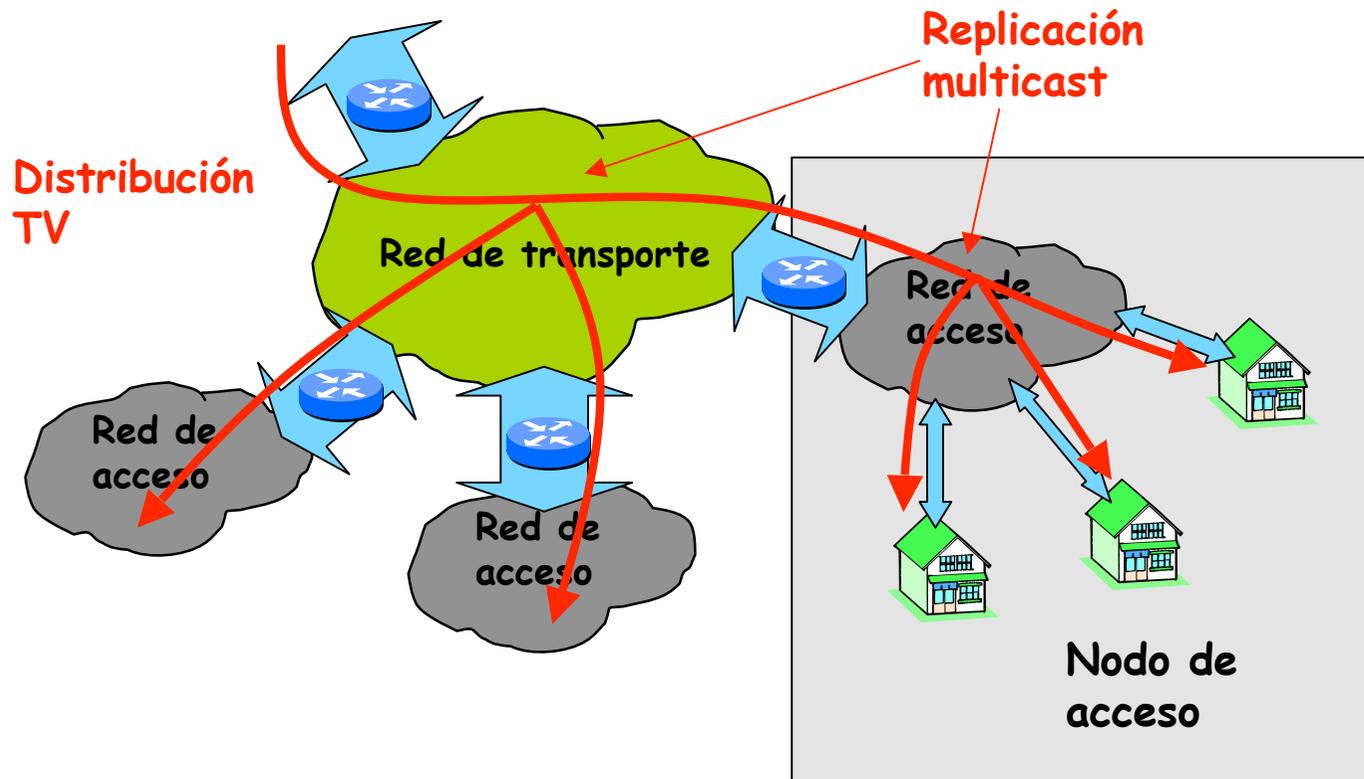
(Imagenio)



# Difusión de TV

## Multicast para difusión de TV

- Escalabilidad para un alto número de canales y un número ilimitado de usuarios
- Uso óptimo del ancho de banda

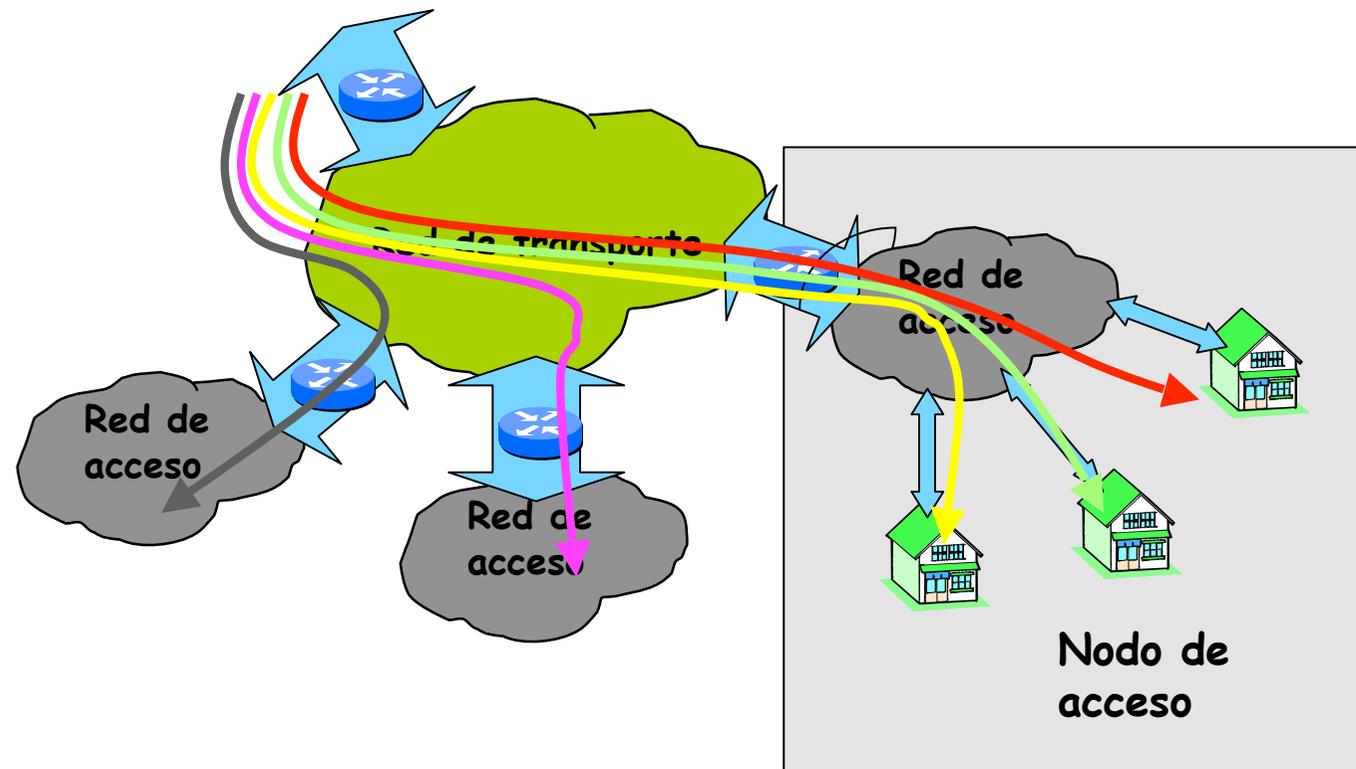




# VoD

## VoD equivale a Unicast

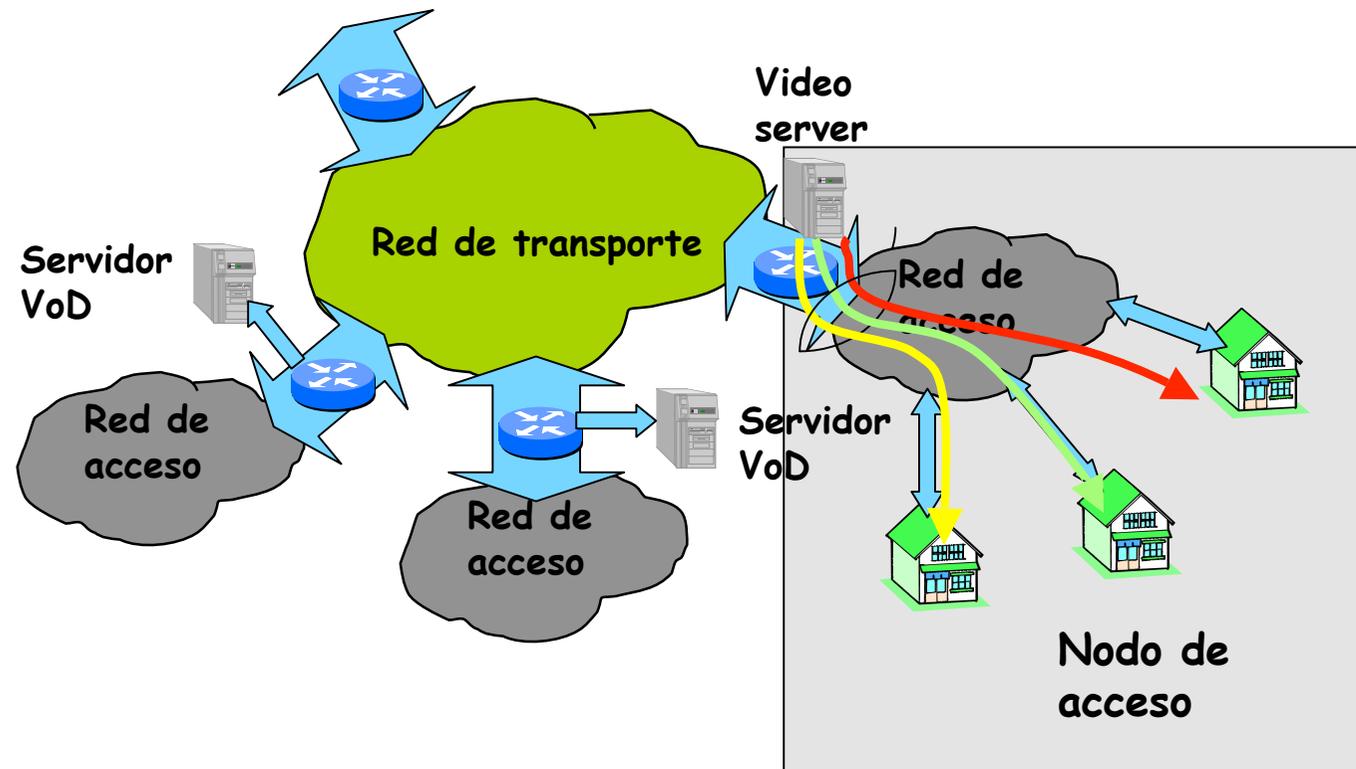
- 1M usuarios @ 2Mb/s & 20% concurrencia = 400Gb/s
- Arquitectura centralizada proporciona escalabilidad limitada



# VoD

## Arquitectura distribuida de VoD

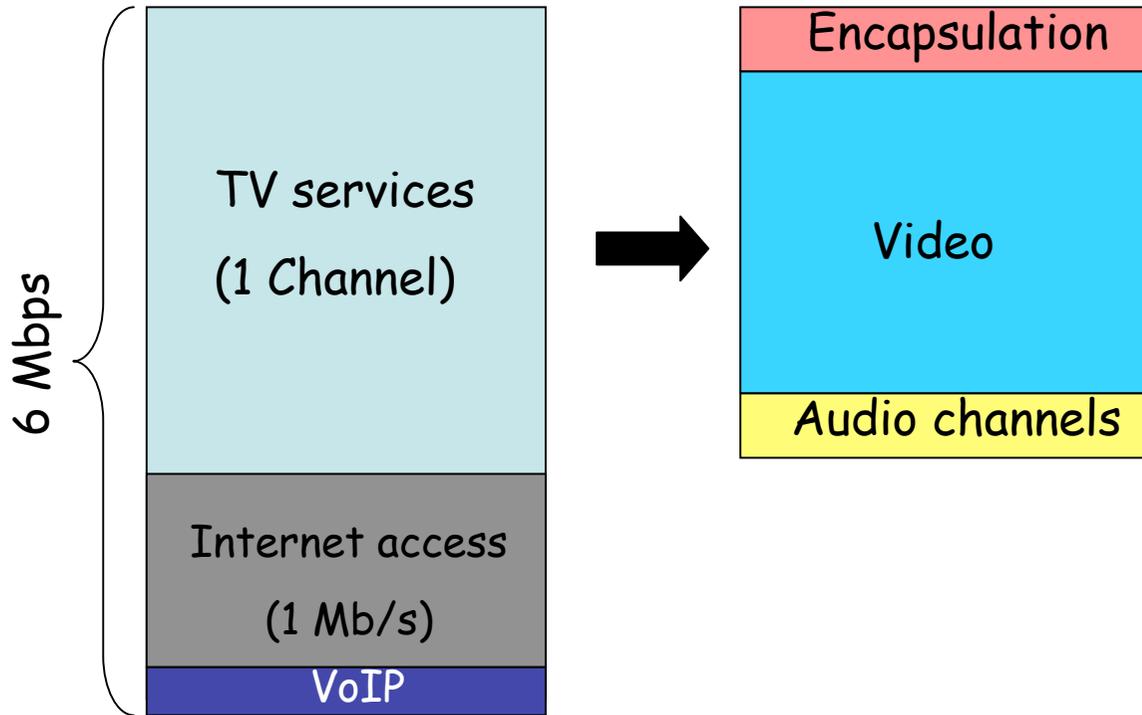
- 20k usuarios @ 2Mb/s & 20% concurrencia = 8Gb/s
- Arquitectura distribuida proporciona escalabilidad
- También es posible una arquitectura mixta (centralizada/distribuida)





# Red de acceso ADSL

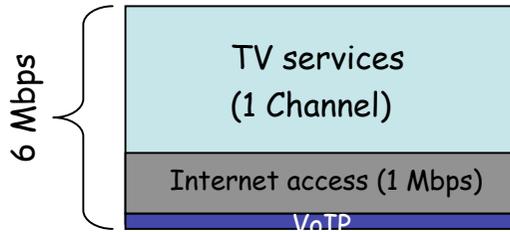
Hoy



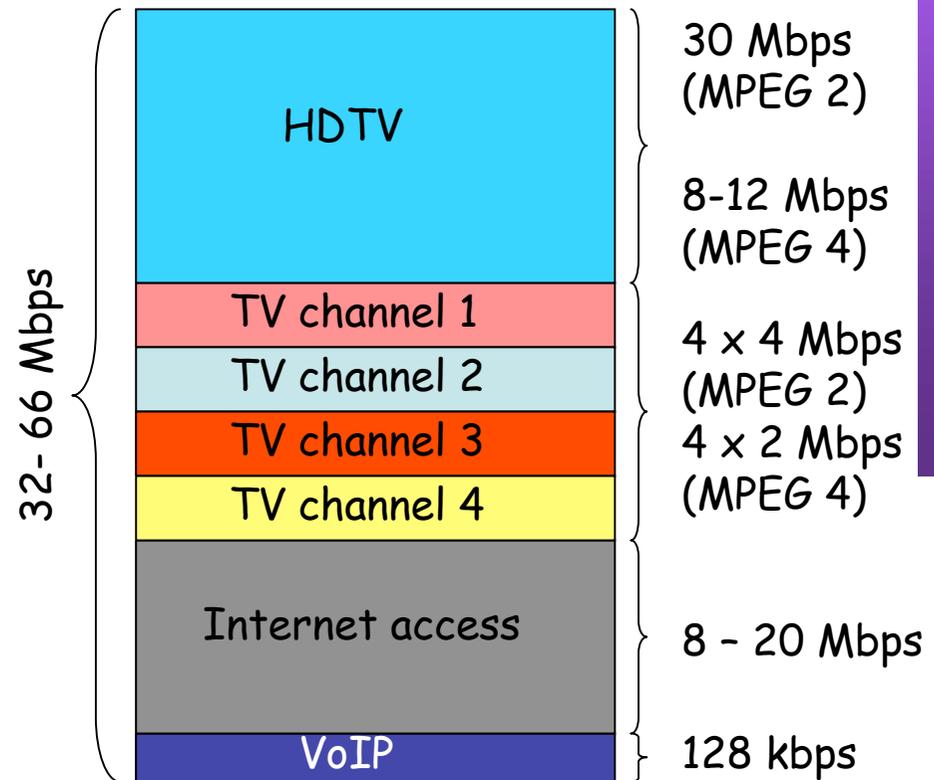


# Red de acceso ADSL2+

Hoy

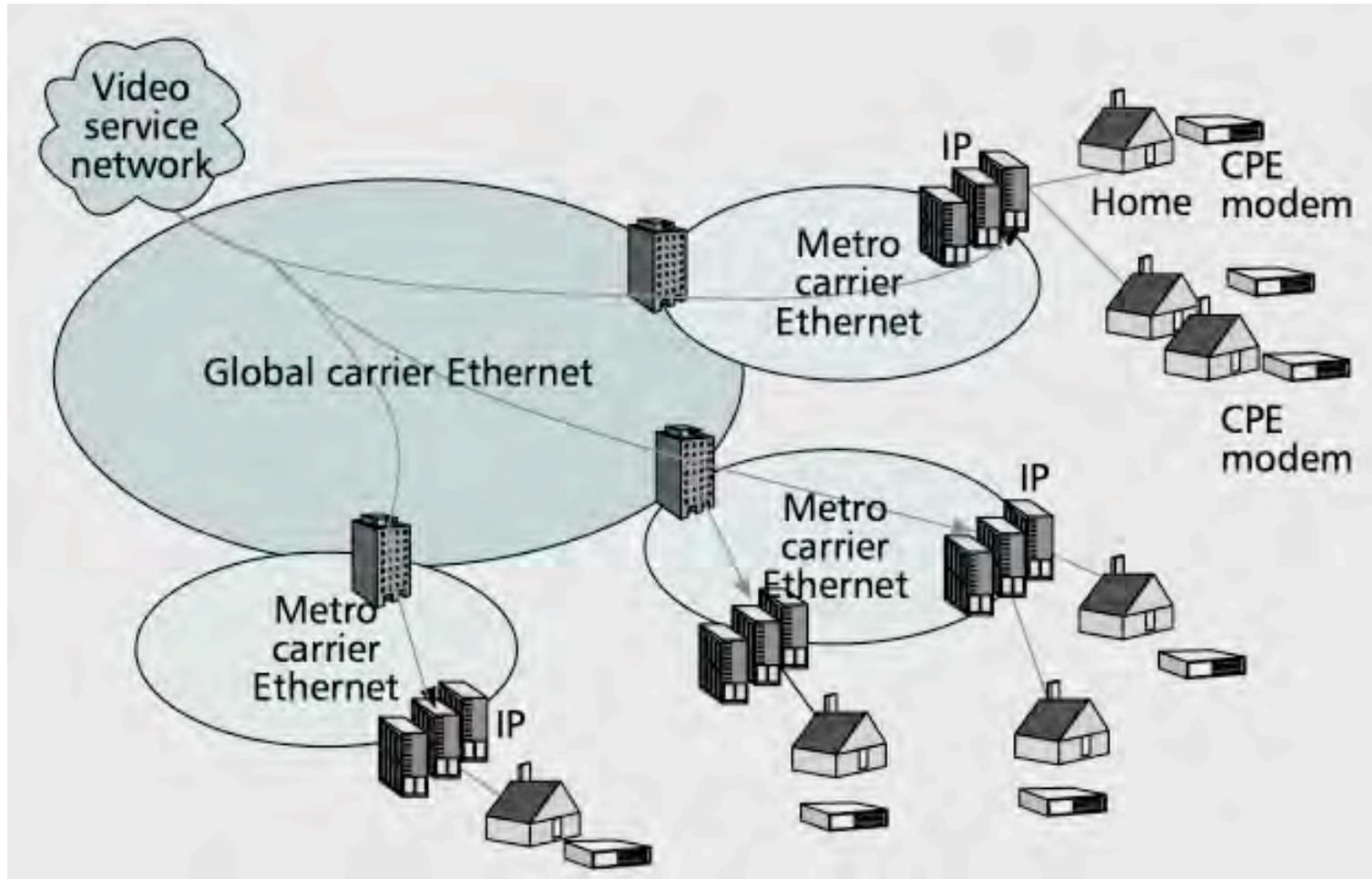


Futuro



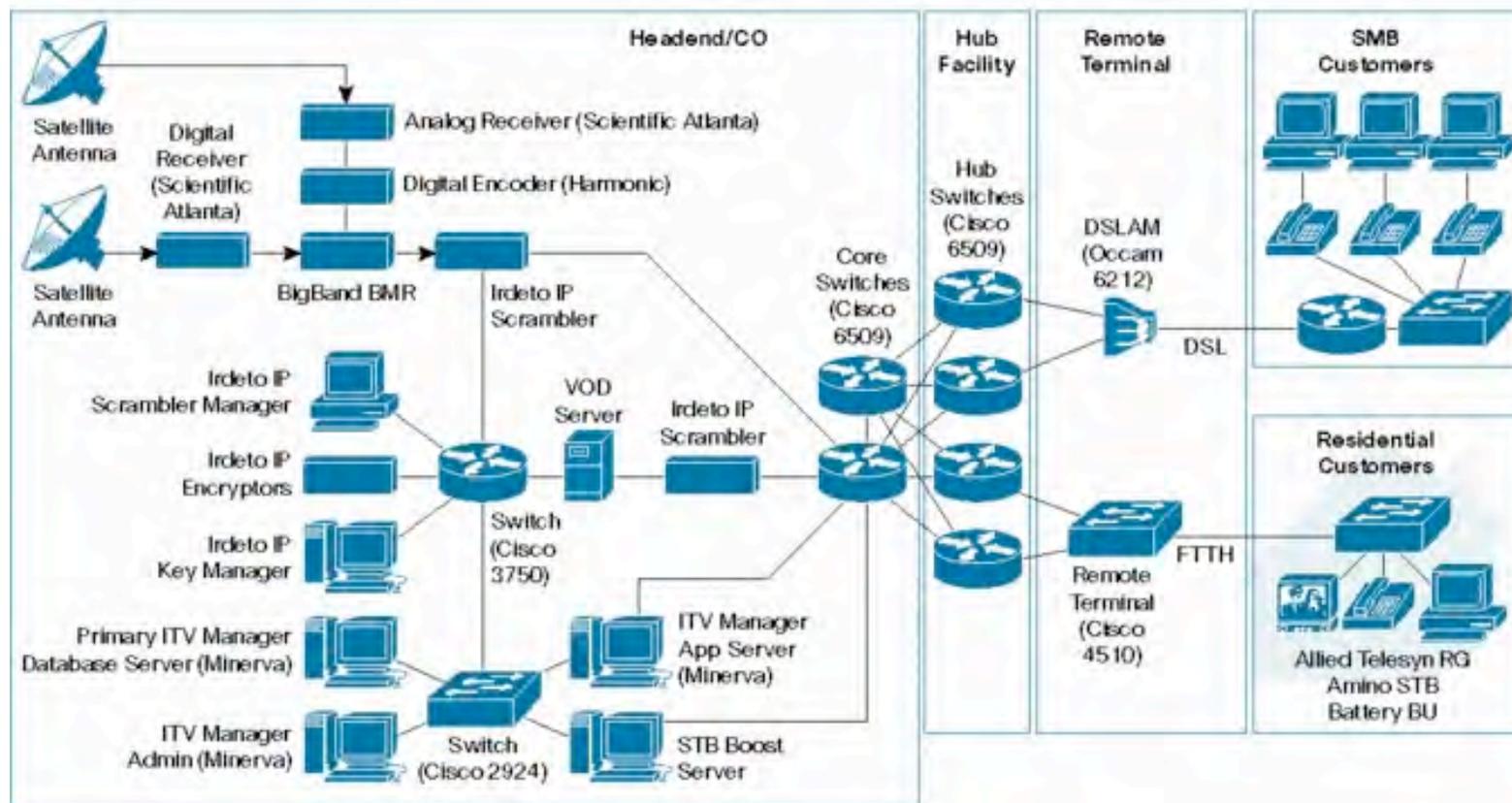


# Arquitecturas de red





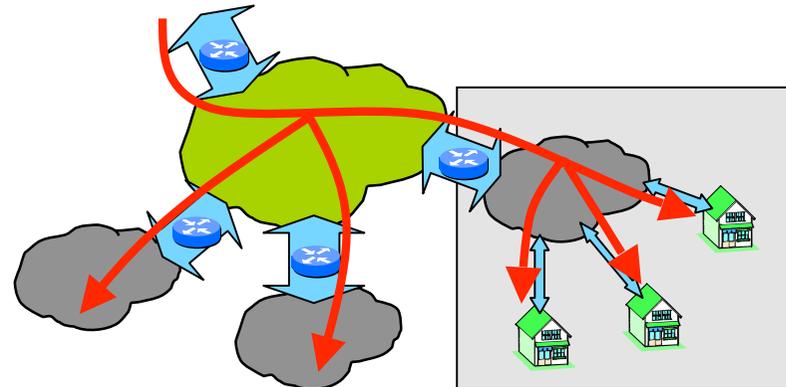
# Arquitecturas de red





# ¿Necesidades?

- Servidor:
  - Escalable ante numerosas peticiones
- Red:
  - Escalabilidad  $\Rightarrow$  Multicast
  - Sin cortes  $\Rightarrow$  QoS (tecnología?)
- Dimensionamiento:
  - ¿Cómo es el tráfico de vídeo?
  - ¿Cómo se comportan los usuarios?





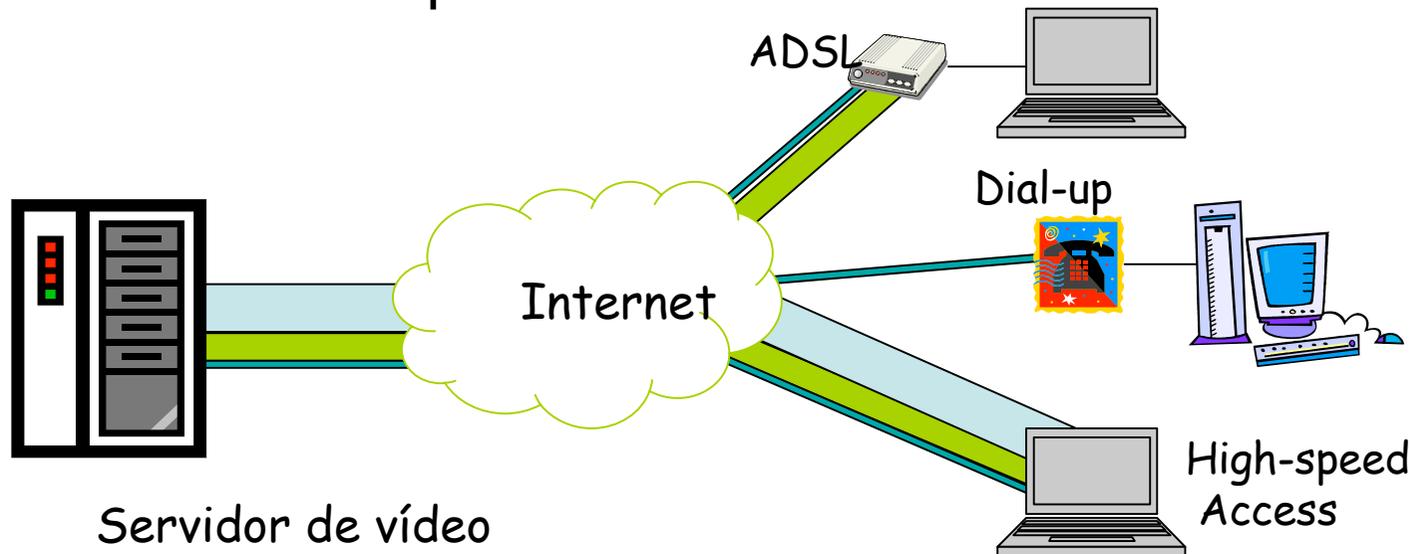
**SERVICIOS EN LA WEB Y DISTRIBUCIÓN DE CONTENIDOS**  
*Área de Ingeniería Telemática*

# El servidor



# Streaming Live Multimedia

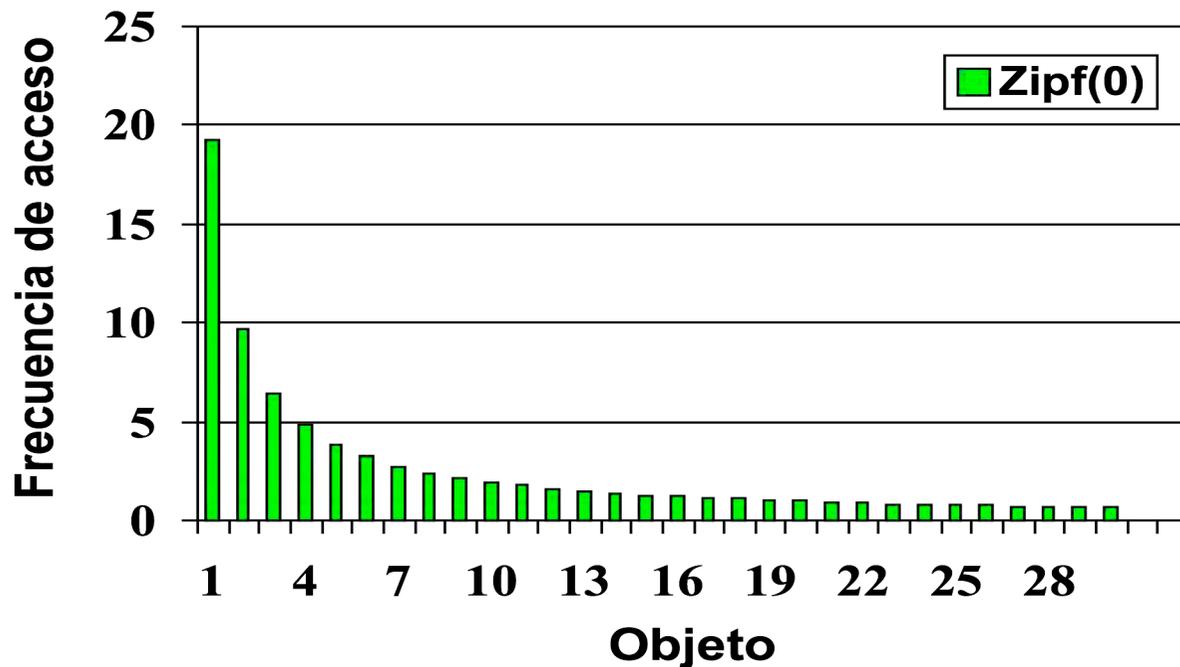
- ¿Cómo hacer *streaming* a un gran número de clientes?
  - Ejemplo: Un evento deportivo popular
  - Usar multicast/broadcast
- ¿Y la heterogeneidad de clientes?
  - Los clientes puede disponer de diferente BW
  - Vídeo en capas





# Comportamiento de los usuarios

- 100s – 1000s solicitudes de un fichero / duración del su reproducción
- Popularidad sesgada de los ficheros
  - 10% – 20% de los ficheros acaparan el 80% de las peticiones
  - Tiene sentido centrarse en los ficheros populares





# Streaming escalable: Motivación

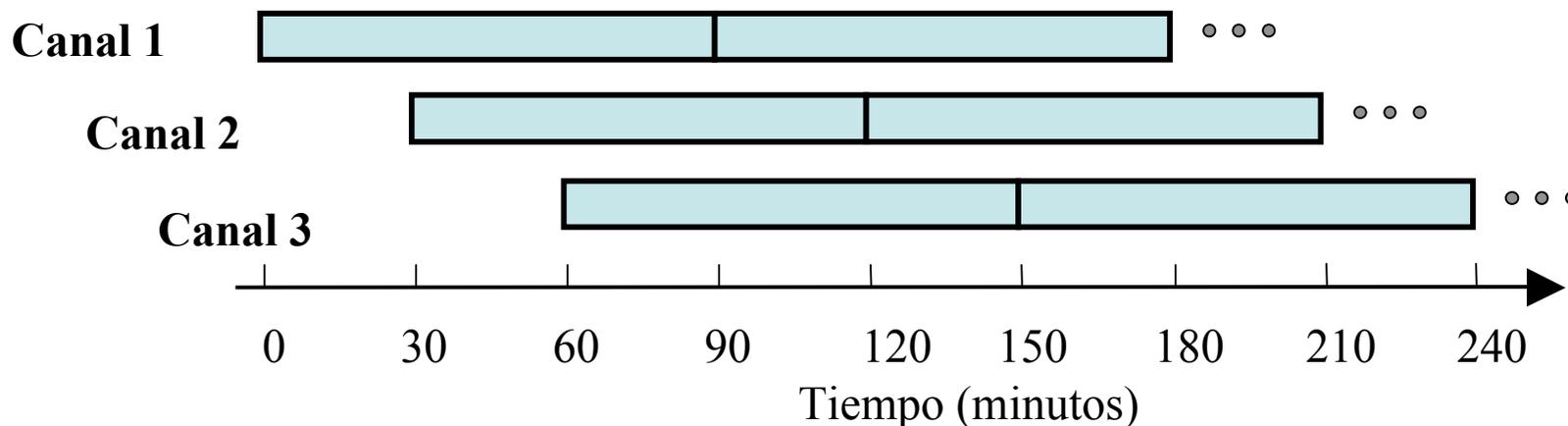
- Consideremos un fichero popular
  - Tasa de reproducción (*playback rate*): 1 Mbps
  - Duración: 90 minutos
  - Peticiones: una por minuto
- Comienza un nuevo stream por cada petición
  - BW requerido = 1 Mbps x 90
- ¿Cómo dar flexibilidad en el instante de comienzo sin enviar un flujo por petición?
- *Batching*
  - Acumular suficientes peticiones para que sea rentable iniciar un nuevo flujo
  - Empezar el flujo al acumular suficientes usuarios o llegar a un límite de tiempo de espera





# Batching (Ejemplo)

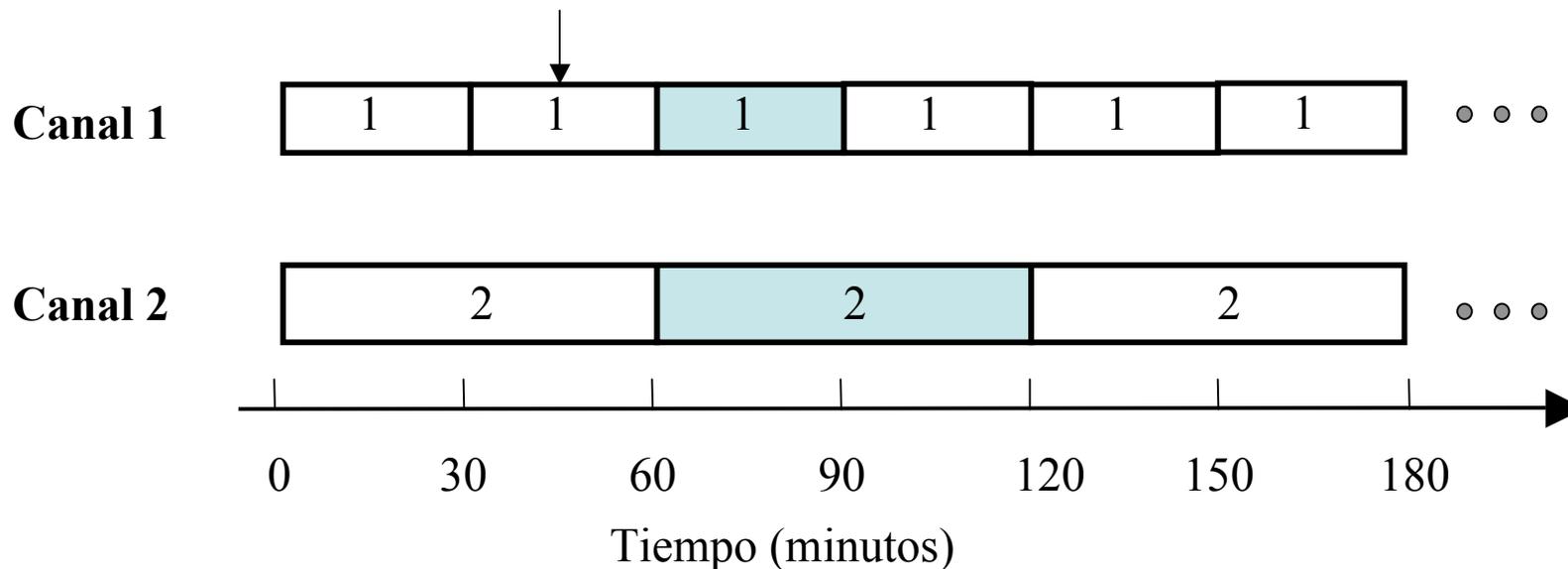
- Playback rate = 1 Mbps, duración = 90 minutes
- Agrupar peticiones en intervalos no solapados de 30 minutos:
  - Máx espera comienzo = 30 minutos
  - BW necesario = 3 canales = 3 Mbps
- BW aumenta linealmente con la reducción de la espera de comienzo





# Periodic Broadcast

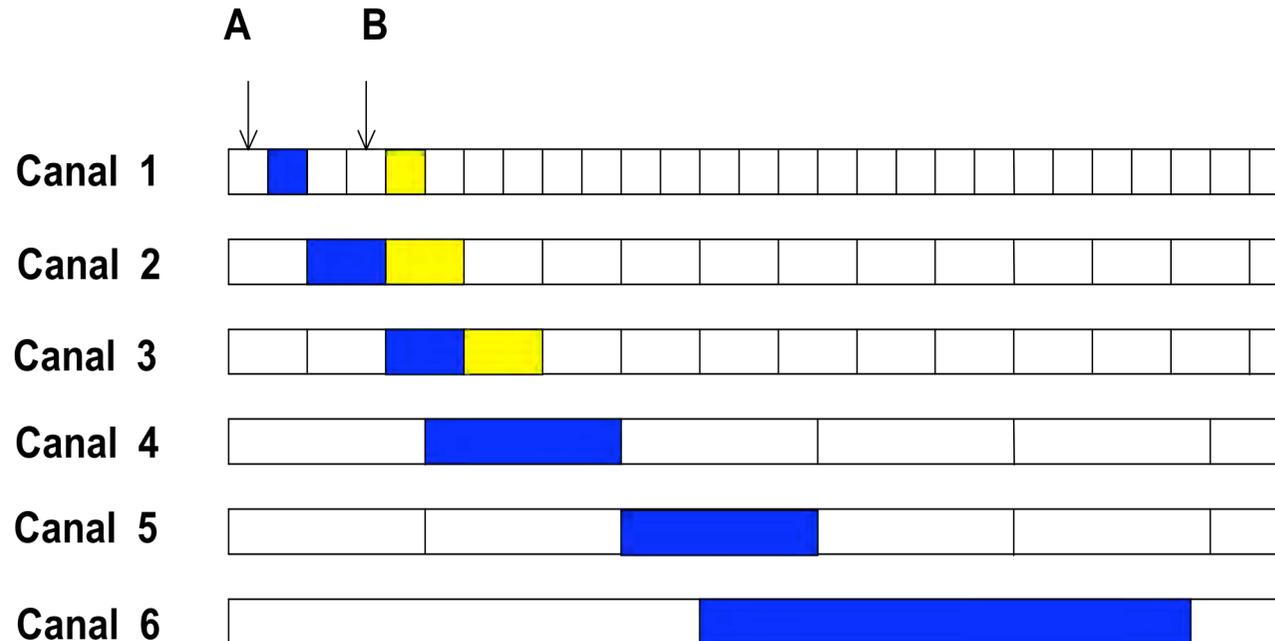
- Partir el fichero en 2 segmentos con tamaños relativos {1,2}
- Para una película de 90 minutos:
  - Segmento 1 = 30 minutos, Segmento 2 = 60 minutos
- Ventaja:
  - Máx espera comienzo = 30 minutos
  - BW necesario = 2 canales = 2 Mbps
- Desventajas: Requiere más del cliente (recibir 2 a la vez)





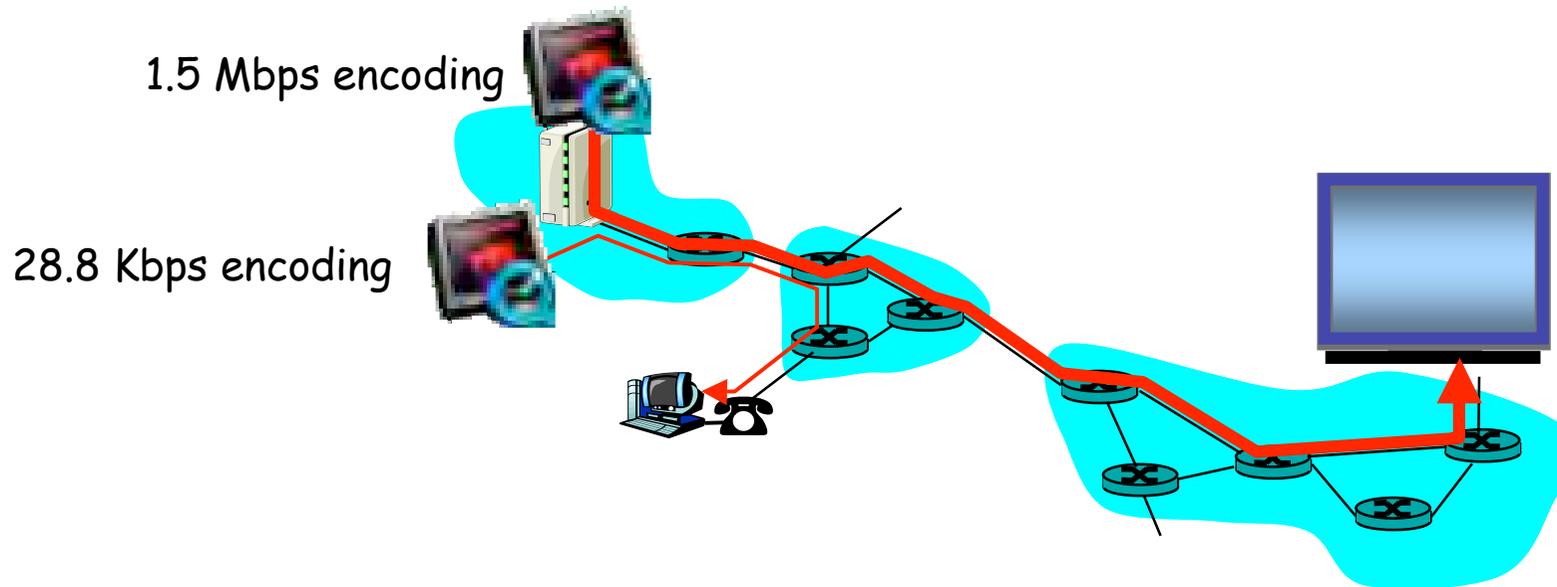
# Skyscraper Broadcasts (SB)

- Divide el fichero en K segmentos de tamaños crecientes
  - Progresión de tamaños: 1, 2, 2, 5, 5, 12, 12, 25, ...
- Multicast de cada segmento en un canal separado
- Tasa a cada cliente: 2 x playback rate





# Streaming: client rate(s)



**Q:** ¿Cómo gestionar diferentes capacidades de recepción?

- 28.8 Kbps modem
- 100Mbps Ethernet

**A:** El servidor almacena y transmite copias del vídeo codificadas a diferentes velocidades



# Streaming: ¿UDP o TCP?

## UDP

- El servidor envía a la velocidad apropiada para la reproducción
- Retardo de comienzo (2-5 segs) para compensar el jitter de la red
- Recuperación de pérdidas: *time permitting*

## TCP

- Puedes entregar a la velocidad de reproducción pero no sabes cuándo lo enviará TCP
- Además puede acumular bytes para formar paquetes más grandes
- Normalmente envía a la máxima velocidad
- Fluctuará debido a los mecanismos de control de flujo y control de congestión
- Retardo de comienzo más largo para poder suavizar el comportamiento de TCP
- HTTP/TCP atraviesa mejor firewalls



# Multimedia en la Internet actual

- TCP/UDP/IP: “best-effort service”
- Sin garantías de retardo o pérdidas
- Los programas emplean técnicas en el nivel de aplicación para mitigar los efectos de retardos y pérdidas
  - Buffers
  - Algoritmos de codificación resistentes ante pérdidas

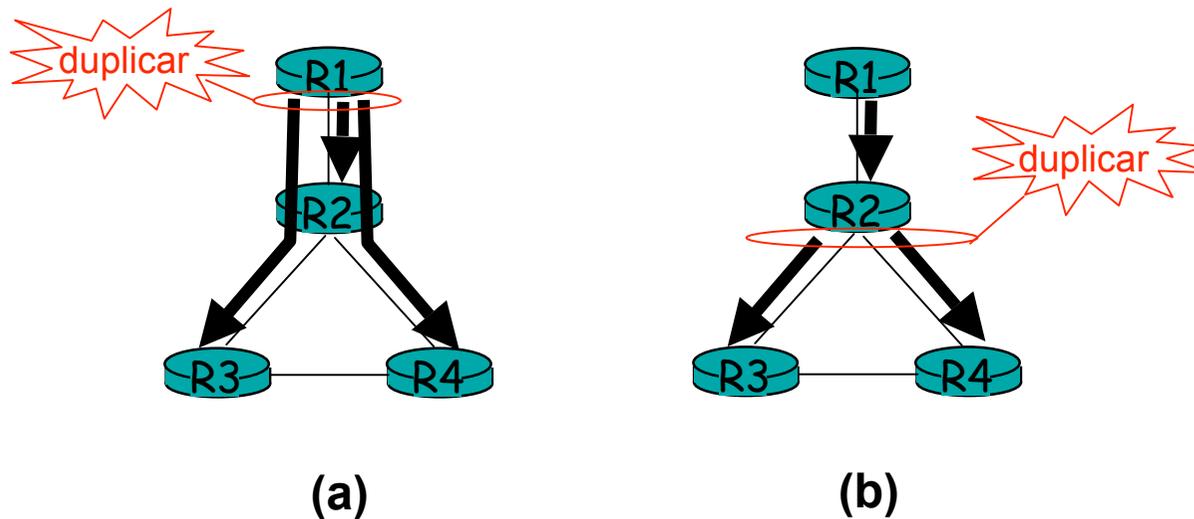


**SERVICIOS EN LA WEB Y DISTRIBUCIÓN DE CONTENIDOS**  
*Área de Ingeniería Telemática*

# Necesidades en la red: Multicast



# Multicast: Introducción



Duplicar en el origen frente a duplicar en la red  
(a) En el origen, (b) en la red



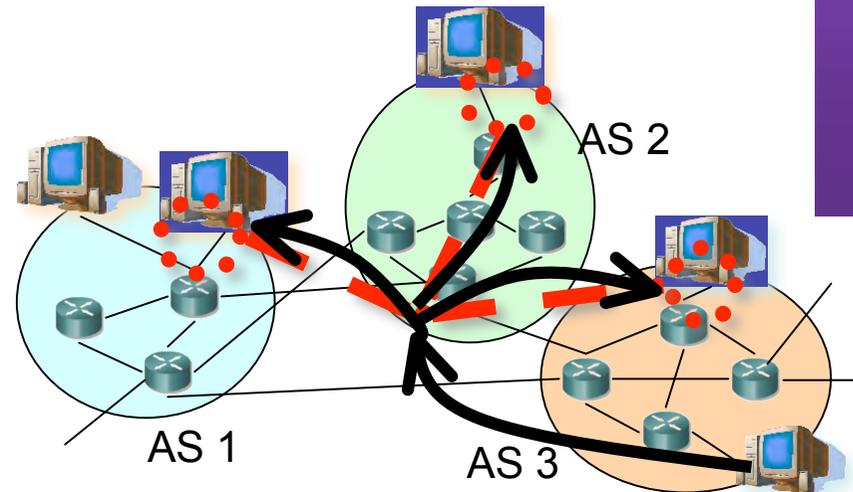
# Multicast

- Grupo abierto
  - Grupo identificado por una dirección independiente de la localización
  - Cualquiera (incluso de fuera del grupo) puede enviar a los miembros del grupo
- Sencillo dentro de LANs que soportan broadcast
- Complicado atravesando redes



# Grupos Multicast

- Los **grupos** pueden ser **permanente o transitorios**
  - Lo que es *permanente* es la dirección del grupo
  - La **pertenencia** a un grupo es **dinámica**
  - No hay límites de localización o número de miembros
  - Un interfaz puede pertenecer a varios grupos
  - No hace falta ser miembro para enviar datagramas al grupo
  - Routers → Mrouter (Multicast Router)
- *¿Cómo se une un host a un grupo?*
  - *¿Cómo llegan los datagramas a todos los miembros del grupo?*



bit 0

1110

Grupo Multicast

bit 31



# Asignación de grupos

## Direcciones multicast controlados por el IANA

<http://www.iana.org/assignments/multicast-addresses>

- 224.0.0/24
  - *Local Network Control Block*
  - Para tráfico de control que no sale de un enlace
  - No debe ser reenviado
  - Ejemplos:
    - 224.0.0.1: All Multicast hosts
    - 224.0.0.2: All routers
    - 224.0.0.5: All OSPF routers
- 224.0.1/24
  - *Internetwork Control Block*
  - Tráfico de control que debe ser reenviado a Internet
  - Ejemplos:
    - 224.0.1.1: NTP
- 224.0.2/24
  - *AD-HOC Block*
  - Para las que no caen en las dos categorías anteriores
- 224.2/16
  - *SDP/SAP Block*
- 223/8
  - *GLOP Block*
  - 223.[ASN]/24

bit 0

1110

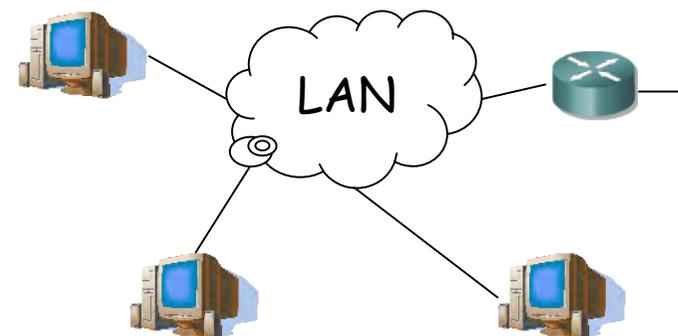
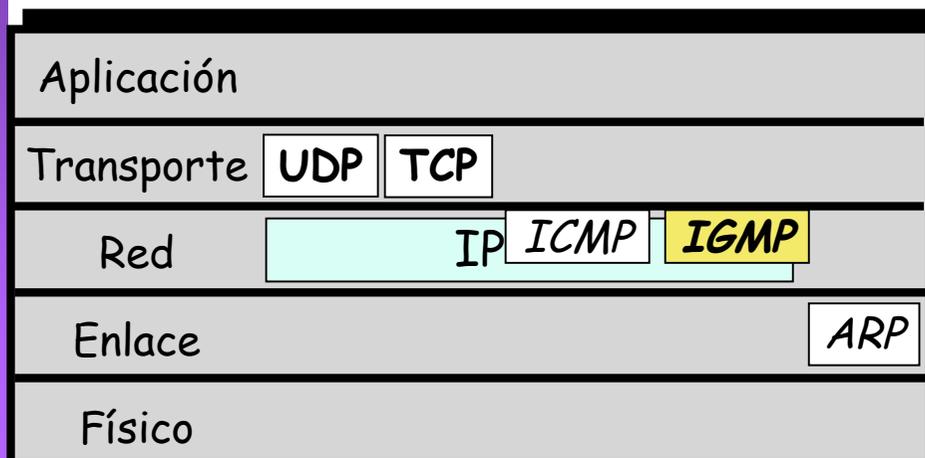
Grupo Multicast

bit 31



# ¿Cómo se une un host a un grupo?

- IGMP = *Internet Group Management Protocol*
- Parte de IP
- Encapsulado en paquetes IP
- *Protocol = 2*
- Entre un host y el router en su red
- Informa de que está interesado en el tráfico de un grupo
- No es entre los hosts
- Hosts no tienen forma de saber qué otros hay en el grupo
- La coordinación entre los routers se realiza mediante otro protocolo
- Todos los interfaces pertenecen al 224.0.0.1 sin anunciarlo





# IGMPv1 (RFC 1112)

## Tipos de mensajes

### 2 = *Report*

- Enviado para unirse al grupo
- IP destino la del grupo
- Enviado desde el interfaz en cuestión
- TTL = 1
- También como respuesta a un *Query*
- En ese caso no responde inmediatamente sino tras un tiempo aleatorio

### 1 = *Query*

- Periódicamente enviado por el router
- Pide actualización de la pertenencia a grupos
- IP destino 224.0.0.1
- TTL = 1
- *Multicast group* a 0s
- Normalmente solo responderá un host de la red

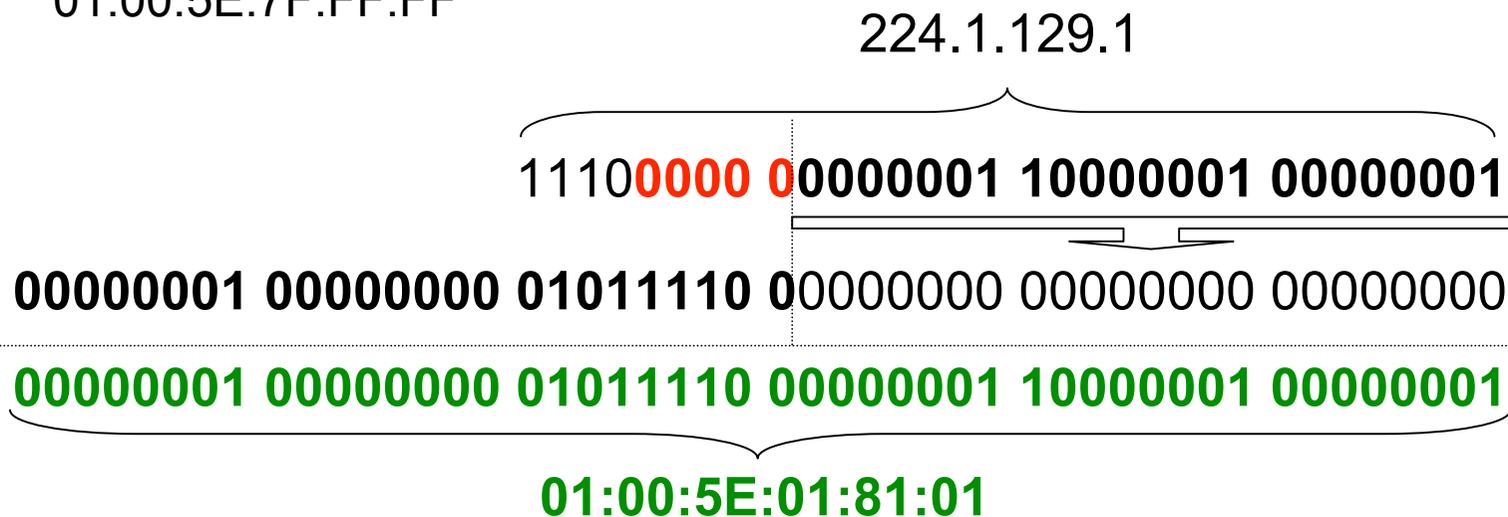
Crear un socket multicast implica el envío automático de un *report*

Version	Type	Unused	Checksum
Multicast group			



# Multicast en LAN Ethernet

- Caso: Paquete multicast a una LAN con hosts miembros
- LAN Ethernet soporta multidifusión
- MACs multicast octavo bit activo
- IANA posee el OID **00:00:5E**
- Reserva la mitad para multicast
- De 01:00:5E:00:00:00 a 01:00:5E:7F:FF:FF
- En los 23 bits bajos se mapean los 23 bits bajos del grupo multicast destino
- 32 grupos colisionan en la misma MAC multicast

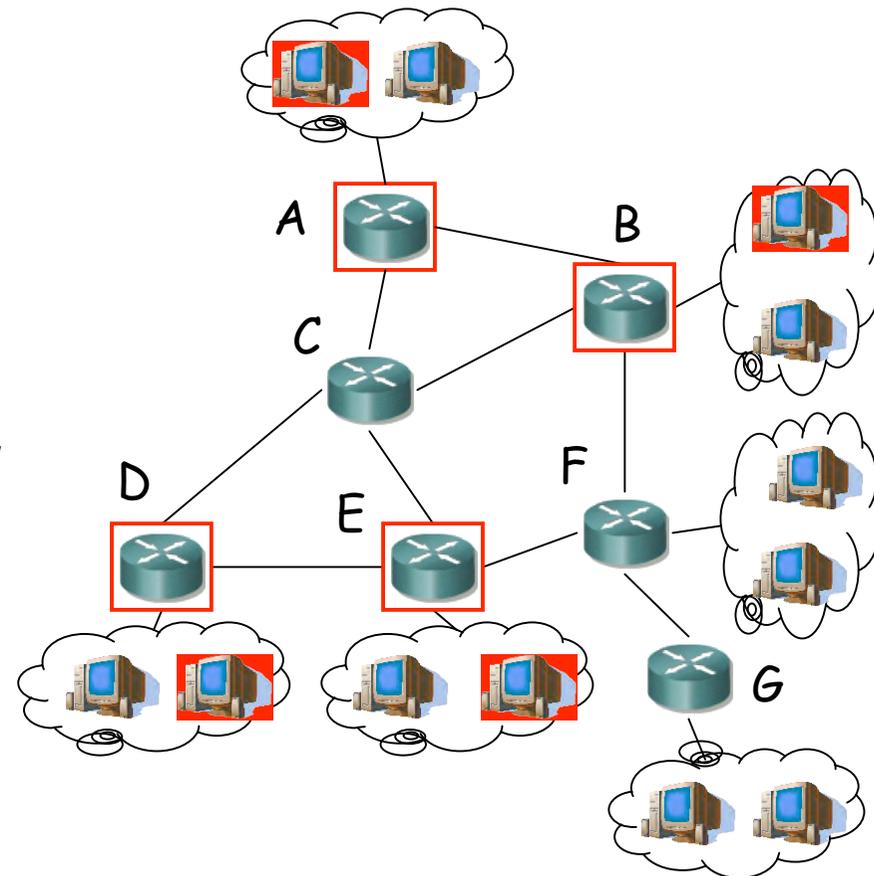




# ¿Cómo llegan los datagramas a todos los miembros del grupo?

## Multicast Routing

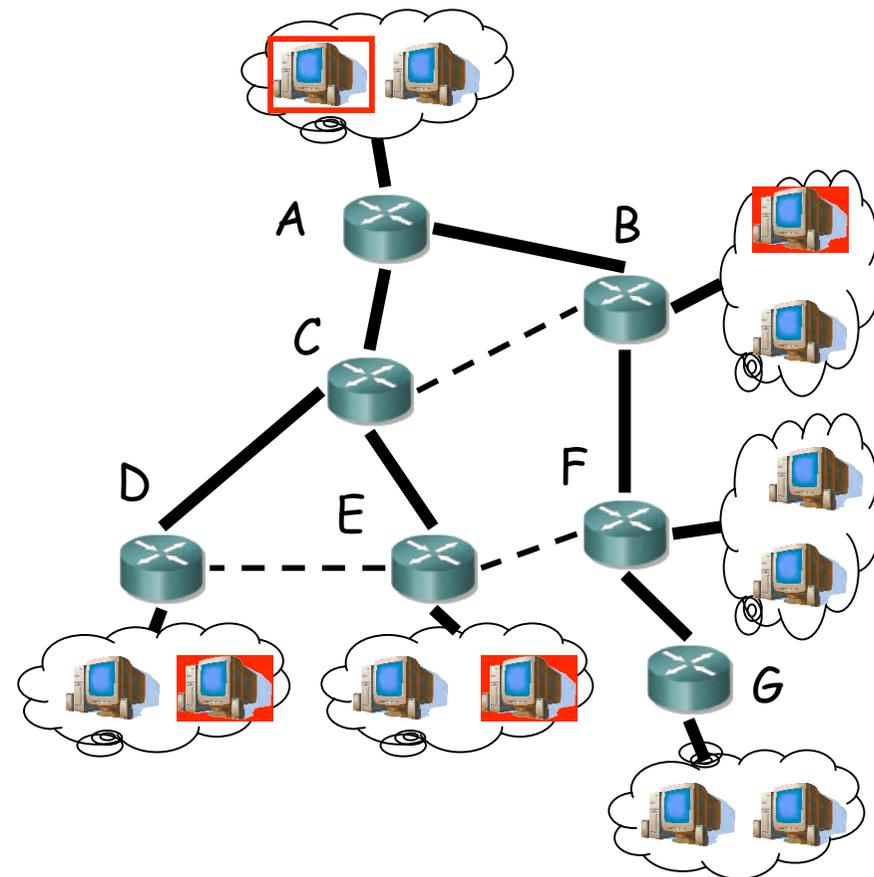
- Encontrar un árbol de enlaces que conecte a todos los routers que sirven a hosts del grupo
- Árboles de expansión (*spanning trees*)
- Mínimos respecto a una métrica (*minimum spanning trees*)
- Puede implicar a otros routers





# Source-Based Trees

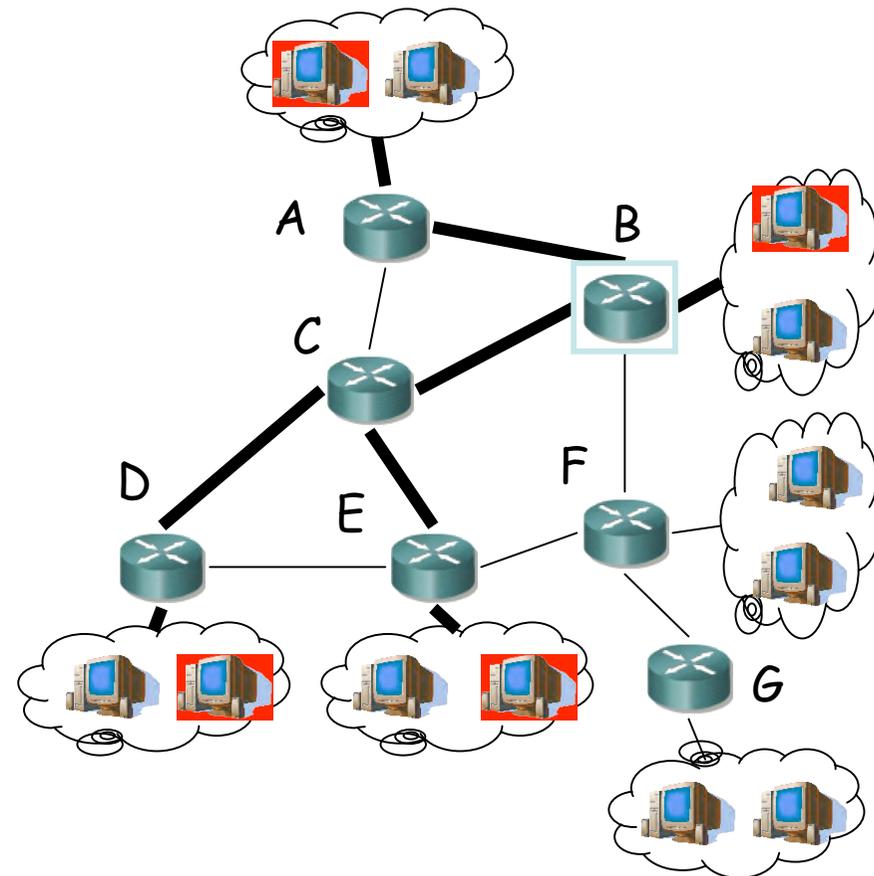
- Un árbol para cada fuente en cada grupo
- $N^{\circ}$  Árboles = Grupos x Fuentes
- Se suele construir empleando *Reverse Path Forwarding (RPF)*:
  - Reenviar por todos los interfaces menos por el que se recibió
  - Solo si se ha recibido por el interfaz por el que se llega al origen por el camino más corto
- **Pruning**
  - MRouter sin hosts ni MRouters adyacentes en el grupo manda mensaje de *prune upstream*
  - Ejemplo: Nodo G





# Group-Shared Tree

- Construir un solo árbol para cada grupo
- Incluye a todos los routers adyacentes a hosts en el grupo
- Se suele construir empleando un **Rendezvous-Point Tree**:
  - Se escoge un nodo central que será la raíz
  - Los demás envían mensajes unicast a él para unirse al árbol
  - Reenviado hasta que el mensaje llega a él o a uno en el árbol

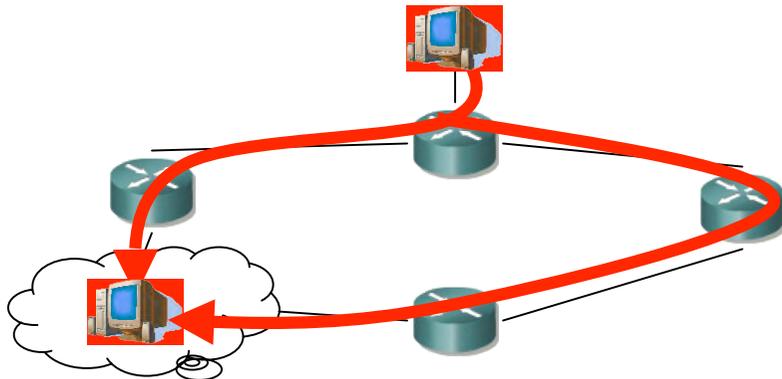




# Algunos protocolos

## DVMRP

- *Distance Vector Multicast Routing Protocol*
- Emplea RPF
- Con RPF puede llegar más de una copia a una red
- **Reverse Path Broadcasting (RPB)**
  - Cada red tiene un *designated parent router*
  - Solo éste router puede reenviar a una red
- Emplea *pruning*
- Emplea nuevos mensajes IGMP



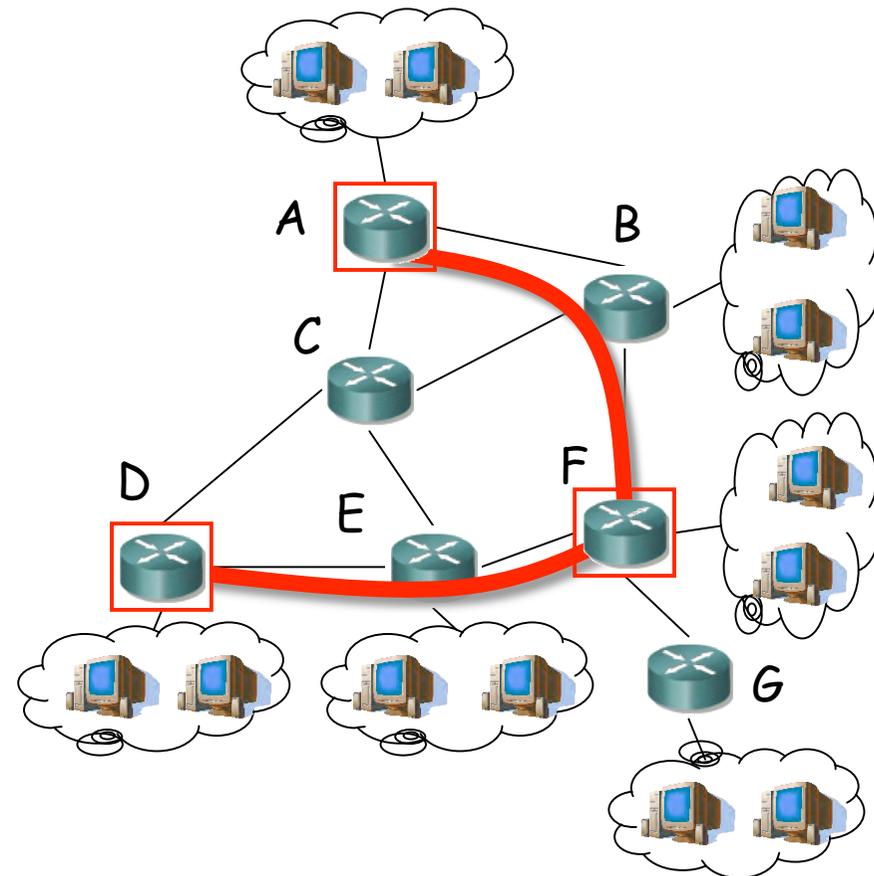
## PIM (*Protocol-Independent Multicast*)

- *Protocol-Independent* porque puede utilizar la información obtenida mediante cualquier protocolo de enrutamiento
- **PIM-DM (Dense Mode)**
  - Cuando casi todos los routers están involucrados
  - Similar a DVMRP
  - *RPF Flood-and-prune*
- **PIM-SM (Sparse Mode)**
  - Pequeño número de routers con hosts adyacentes en el grupo
  - *Group-shared*
- **Otros:** MOSPF, MBGP, CBT...



# MBONE (Túneling)

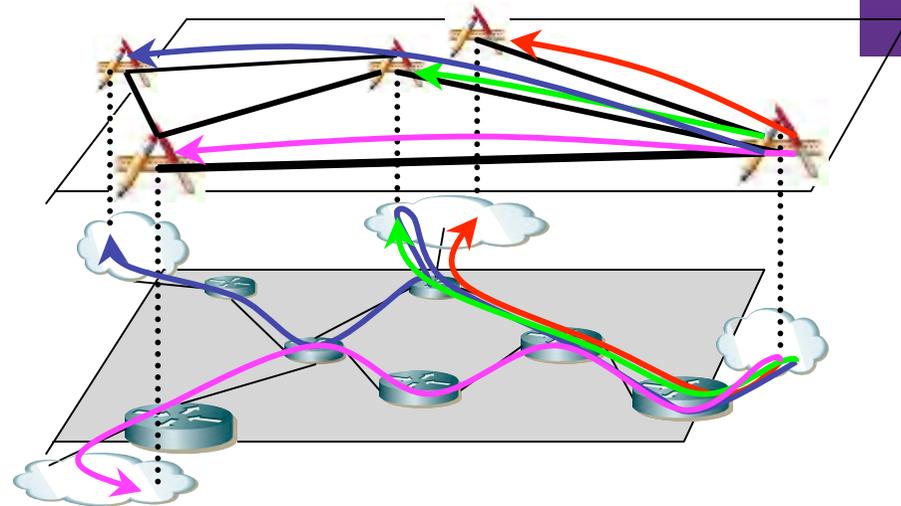
- *Multicast Backbone*
- Partiendo de una red en la que muchos routers no soportan multicast
- MRouters conectados entre sí con **túneles**
- Los mrouter han sido muchas veces estaciones UNIX
- DVMRP
- Session Description Protocol (SDP) para describir las sesiones
- Session Announcement Protocol (SAP) el protocolo de transporte empleado por SDP





# Application layer multicast

- Multicast en el nivel de aplicación
  - Asumiendo que la red subyacente ofrece solo facilidades unicast
  - Los sistemas finales (aplicaciones en los hosts) se comunican a través de una *overlay*
- Ventajas
  - No hay que cambiar los routers
  - Permite incorporar con facilidad nuevas funcionalidades
- Problemas
  - ¿Cómo pueden cooperar para construir una buena estructura overlay?
  - Serias implicaciones en las prestaciones por no optimizar tráfico





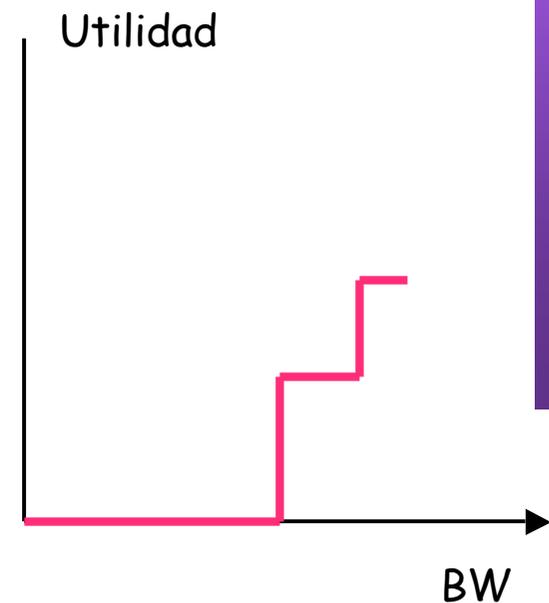
**SERVICIOS EN LA WEB Y DISTRIBUCIÓN DE CONTENIDOS**  
*Área de Ingeniería Telemática*

# Necesidades en la red: QoS



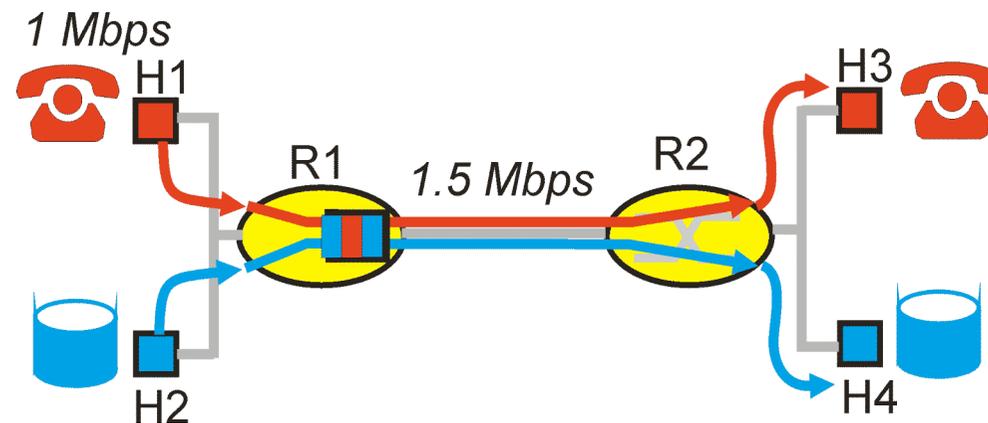
# QoS

- Aplicaciones son sensibles a:
  - Retardos
  - Pérdidas
- Por debajo de un umbral puede no ser útil el tráfico
- Servicio original en Internet *Best Effort*
  - Se trata igual a todo el tráfico (FIFO)
  - No hay mecanismos para distinguir al tráfico *delay-sensitive*



# Principio 1: Clasificación

- Ejemplo: Teléfono IP a 1Mbps, comparte enlace de 1.5Mbps con FTP
  - Ráfagas de FTP pueden congestionar el router y causar fallos en el audio
  - Queremos dar prioridad al audio sobre el FTP

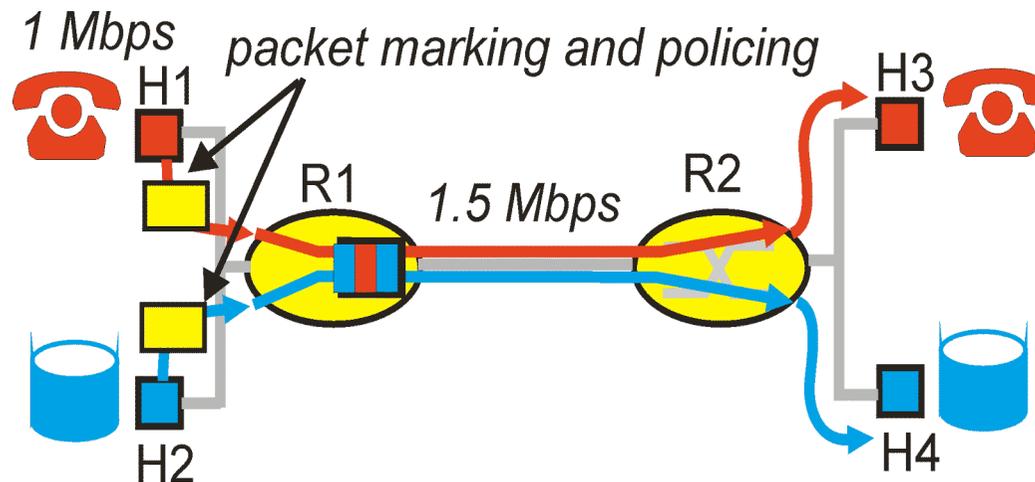


## Principio 1

Los routers necesitan distinguir el tráfico de diferentes clases y aplicarles diferentes políticas: *packet marking* (generalmente a la entrada a la red)

# Principio 2: Aislamiento

- ¿Qué sucede si las aplicaciones no se comportan como deben?
  - Por ejemplo la aplicación de audio envía más de lo previsto
  - Necesitamos forzar que las fuentes se comporten como se ha acordado

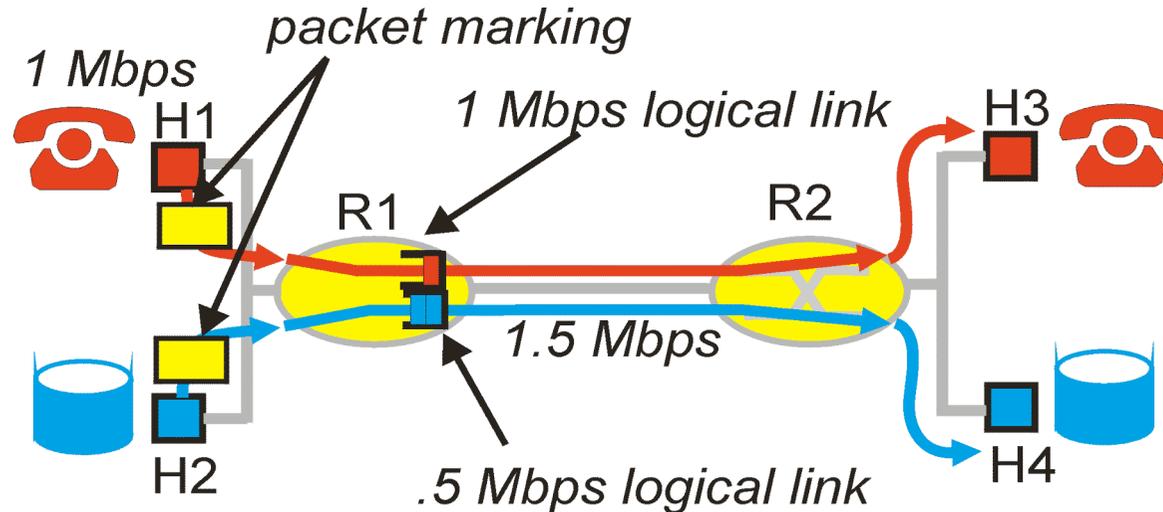


## Principio 2

Forzar que una clase de tráfico se comporte dentro de lo contratado: *policing* (típicamente a la entrada)

# Principio 3: Eficiencia

- Reservar BW fijo (no compartido) a un flujo es ineficiente si ese flujo no lo emplea todo



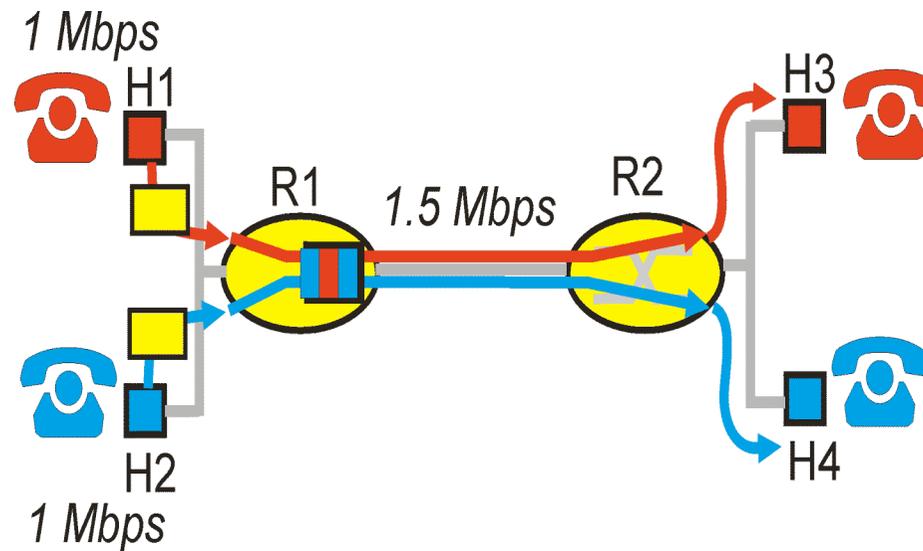
## Principio 3

Mientras se ofrece aislamiento es deseable emplear los recursos de forma eficiente (*work conserving*): *scheduling* (en todos los routers del camino)



# Principio 4: Límite de recursos

- No se pueden satisfacer las demandas más allá de la capacidad del enlace



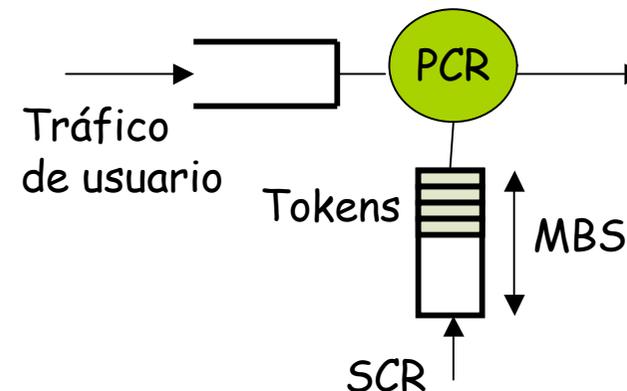
## Principio 4

El flujo declara sus necesidades pero la red puede *bloquear* al flujo si no puede satisfacerlas: *call admission*



# Policing

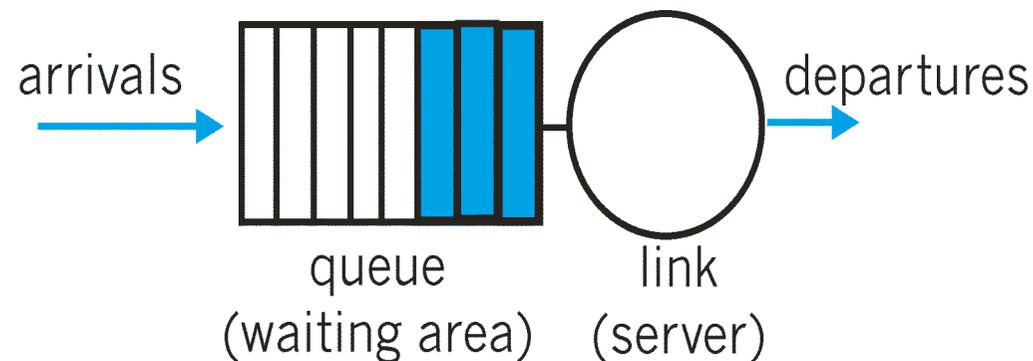
- **Objetivo:** Limitar el tráfico a la entrada a la red para que no exceda el declarado
- Características del tráfico
  - Tasa media: número de paquetes que puede enviar por unidad de tiempo, media a largo plazo
  - Tasa de pico
  - Tamaño máximo de ráfaga: máximo número de paquetes enviados a la tasa de pico
- Los que excedan se descartan
- *Traffic Shaping*: no se descartan sino que se encolan





# Scheduling y Policing

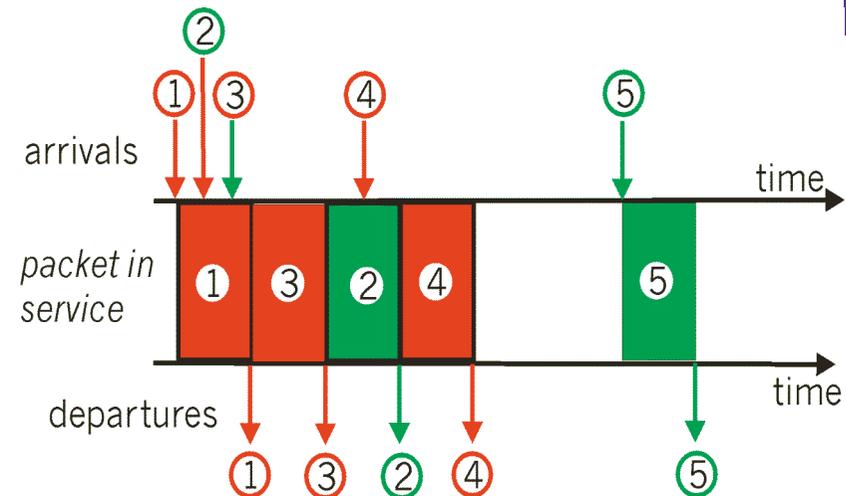
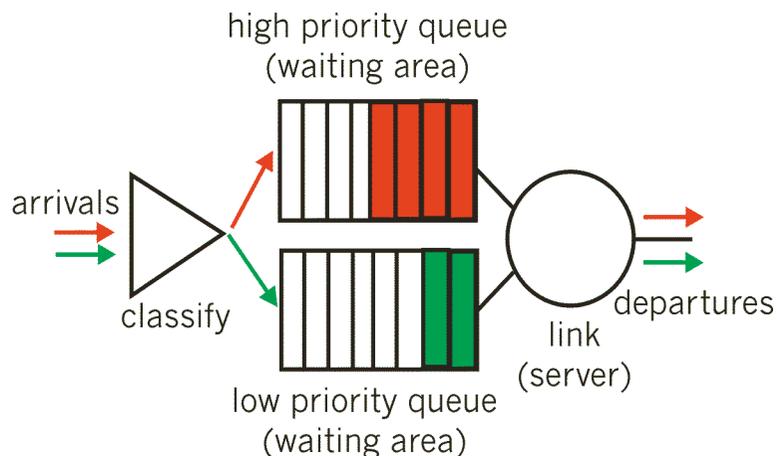
- *Scheduling*: elegir el siguiente paquete a enviar por un enlace
- FIFO (first in first out) scheduling: enviar en orden de llegada
  - *Discard policy*: si un paquete llega y encuentra la cola llena, cuál descartar?
    - *Tail drop*: el que llega
    - *Priority*: Según prioridades
    - *Random*





# Scheduling Policies

- *Priority scheduling*: transmitir el paquete encolado de mayor prioridad
- Múltiples clases, con diferentes prioridades
  - La clase puede depender de etiquetado u otra información en la cabecera (direcciones origen y destino)

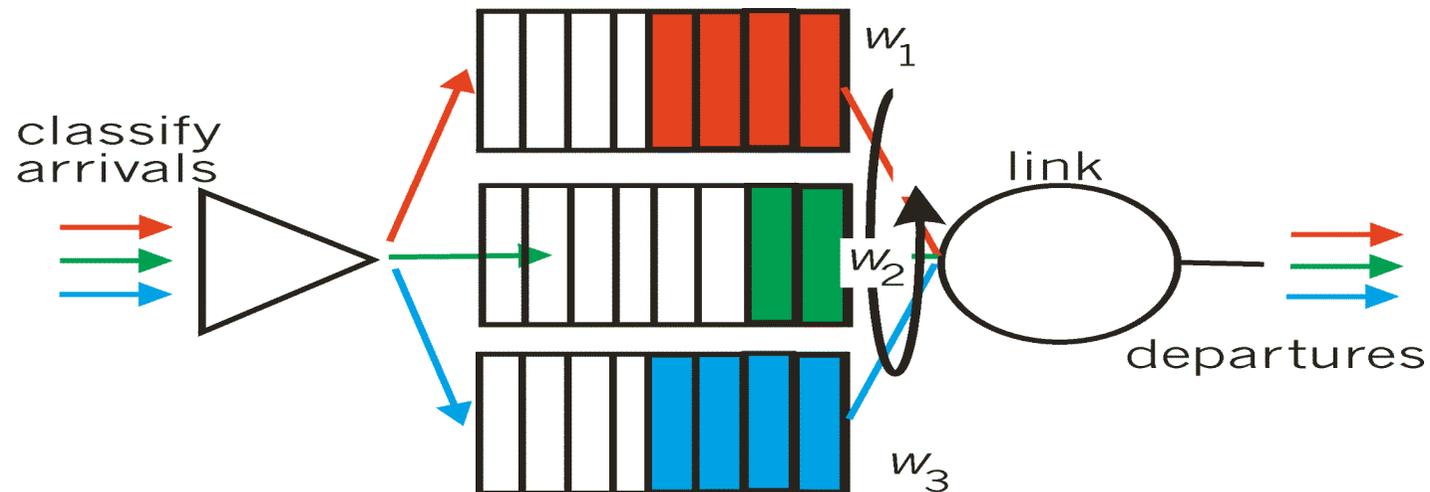




# Scheduling Policies

## Weighted Fair Queuing:

- Cada clase obtiene en cada ciclo un servicio proporcional al peso asignado





**SERVICIOS EN LA WEB Y DISTRIBUCIÓN DE CONTENIDOS**  
*Área de Ingeniería Telemática*

# Tecnologías y QoS



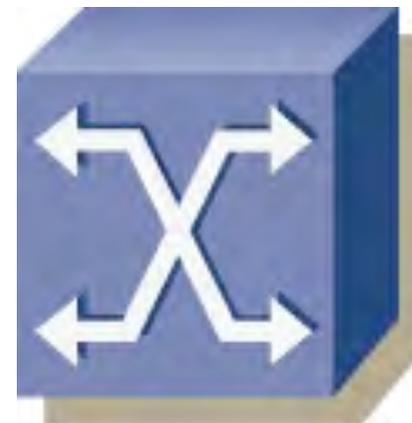
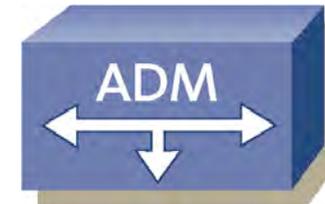
# Tecnologías y QoS

## Mundo de las operadoras:

- (Tecnologías de transporte de datos)
- PDH
- Frame Relay
- SONET/SDH
- ATM

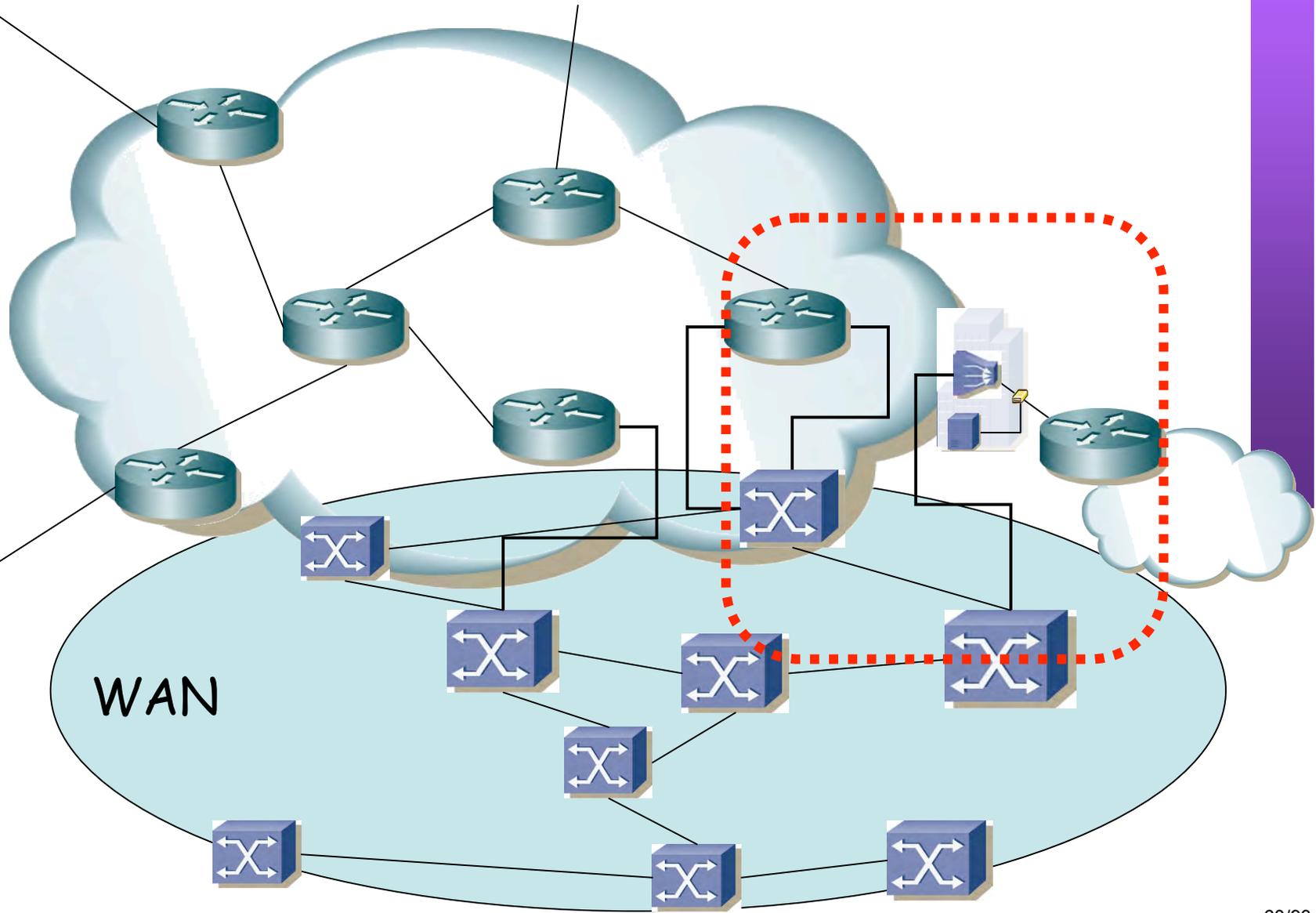
## Internet:

- (IP, que se transportará sobre algo)
- IntServ
- DiffServ
- MPLS





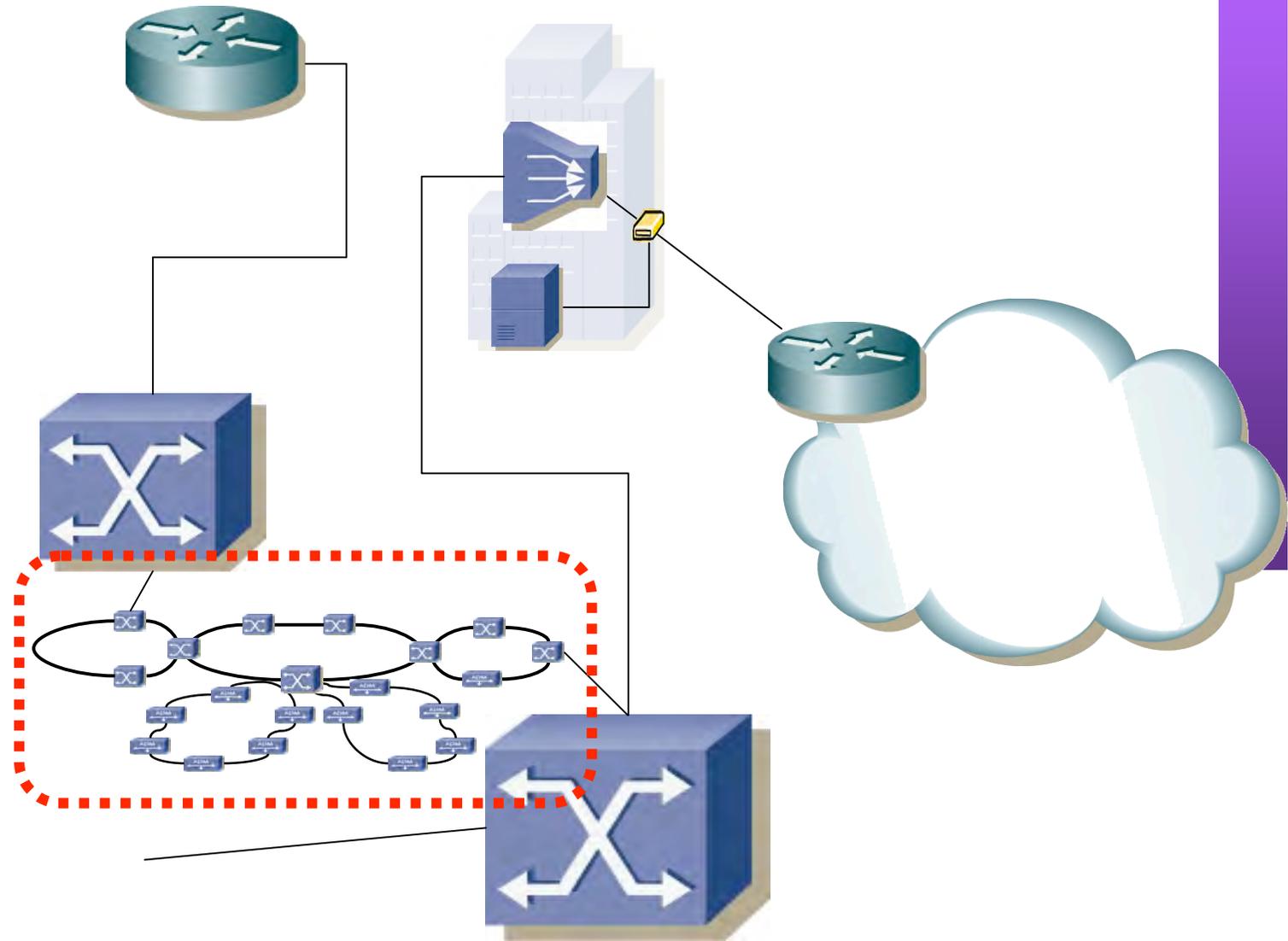
# Tecnologías WAN





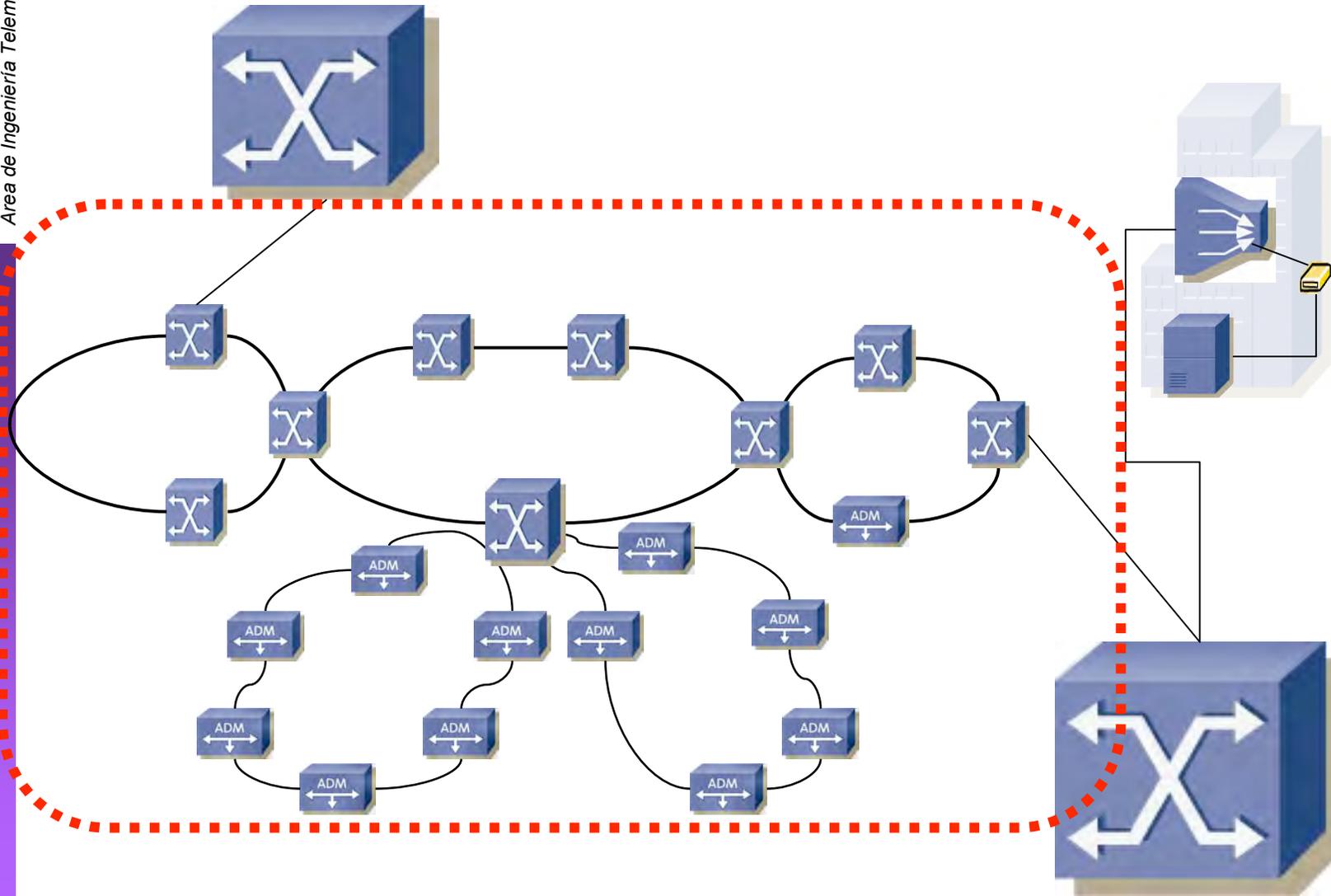


# WAN



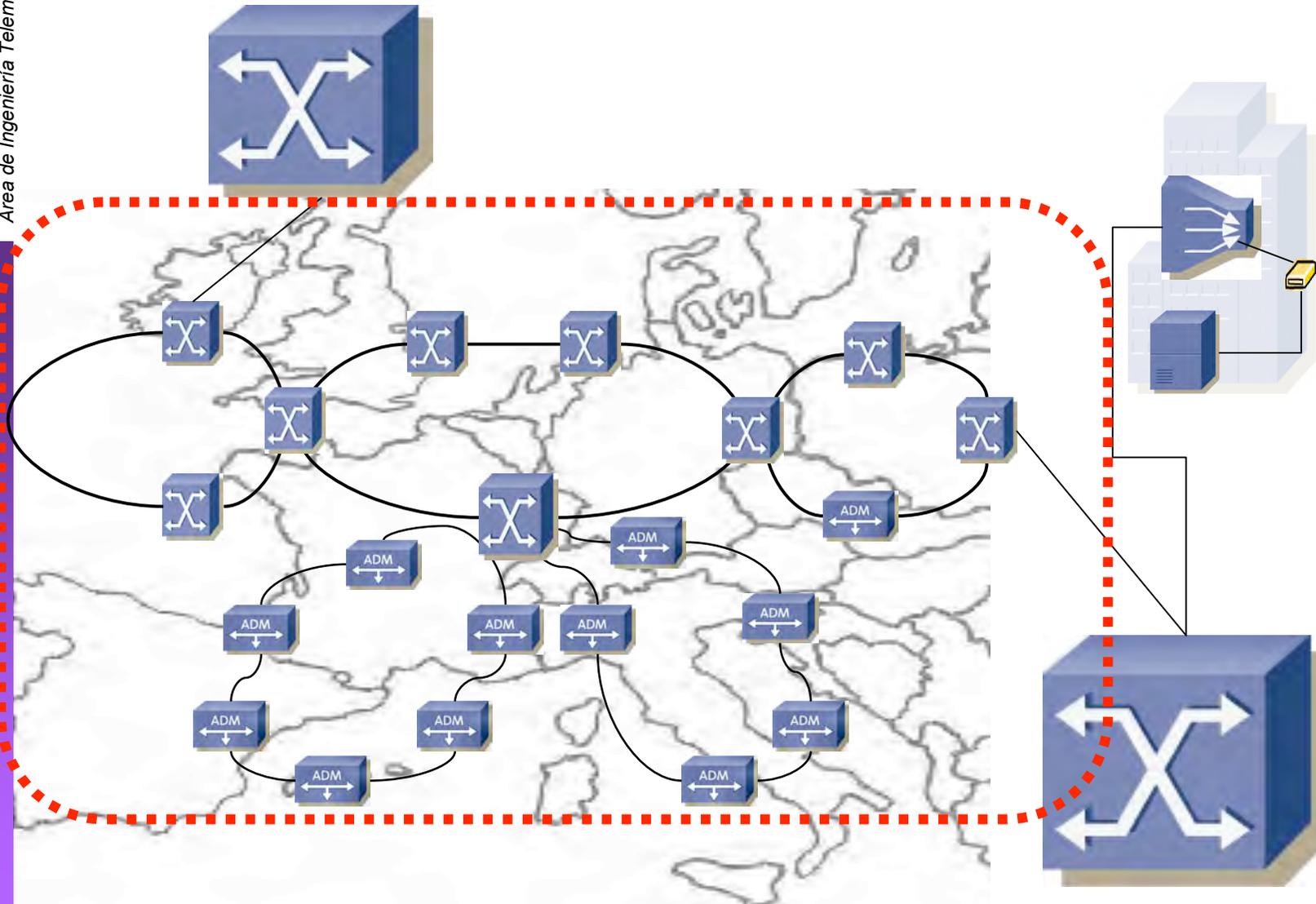


# WAN SDH





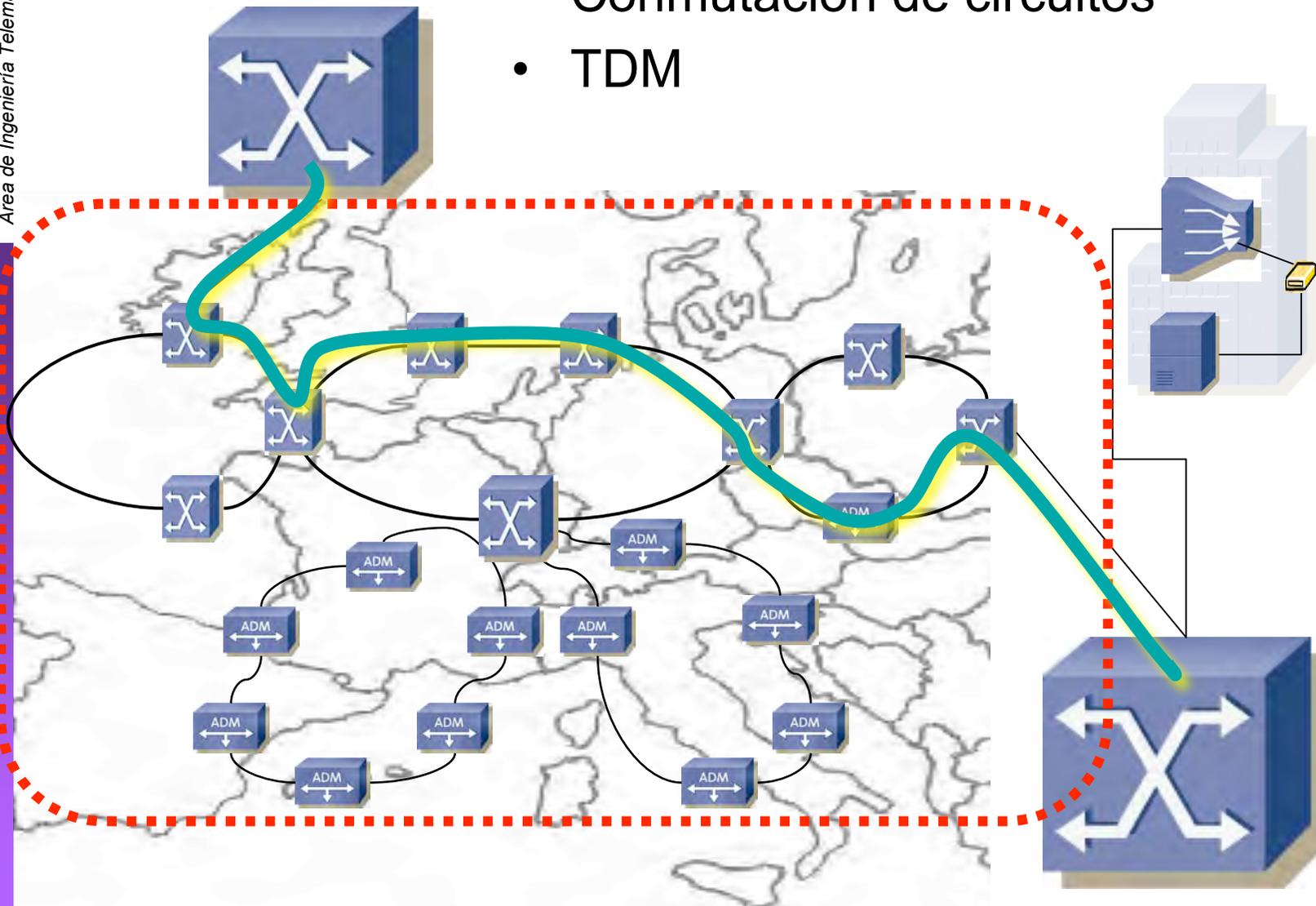
# WAN SDH





# WAN SDH

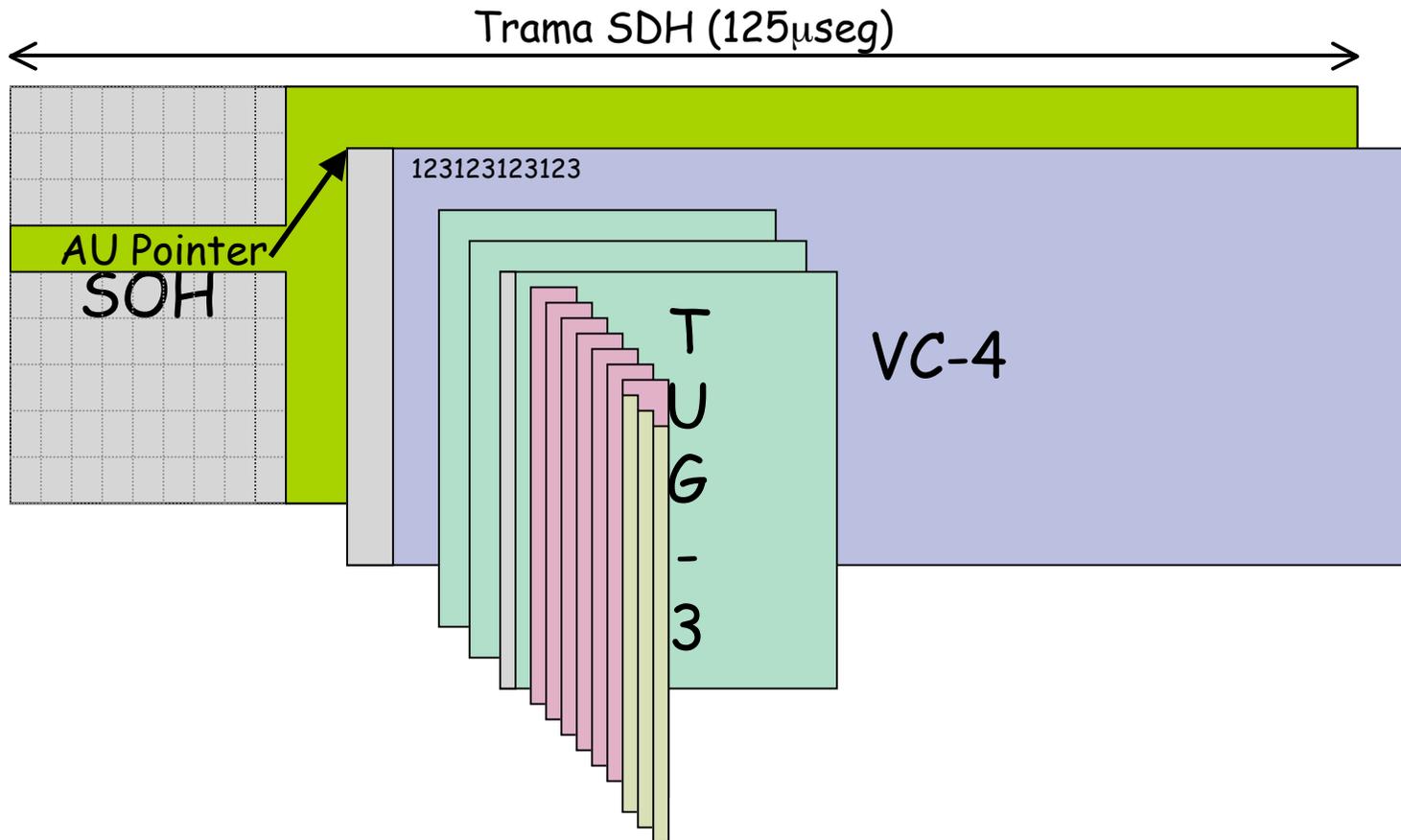
- Conmutación de circuitos
- TDM





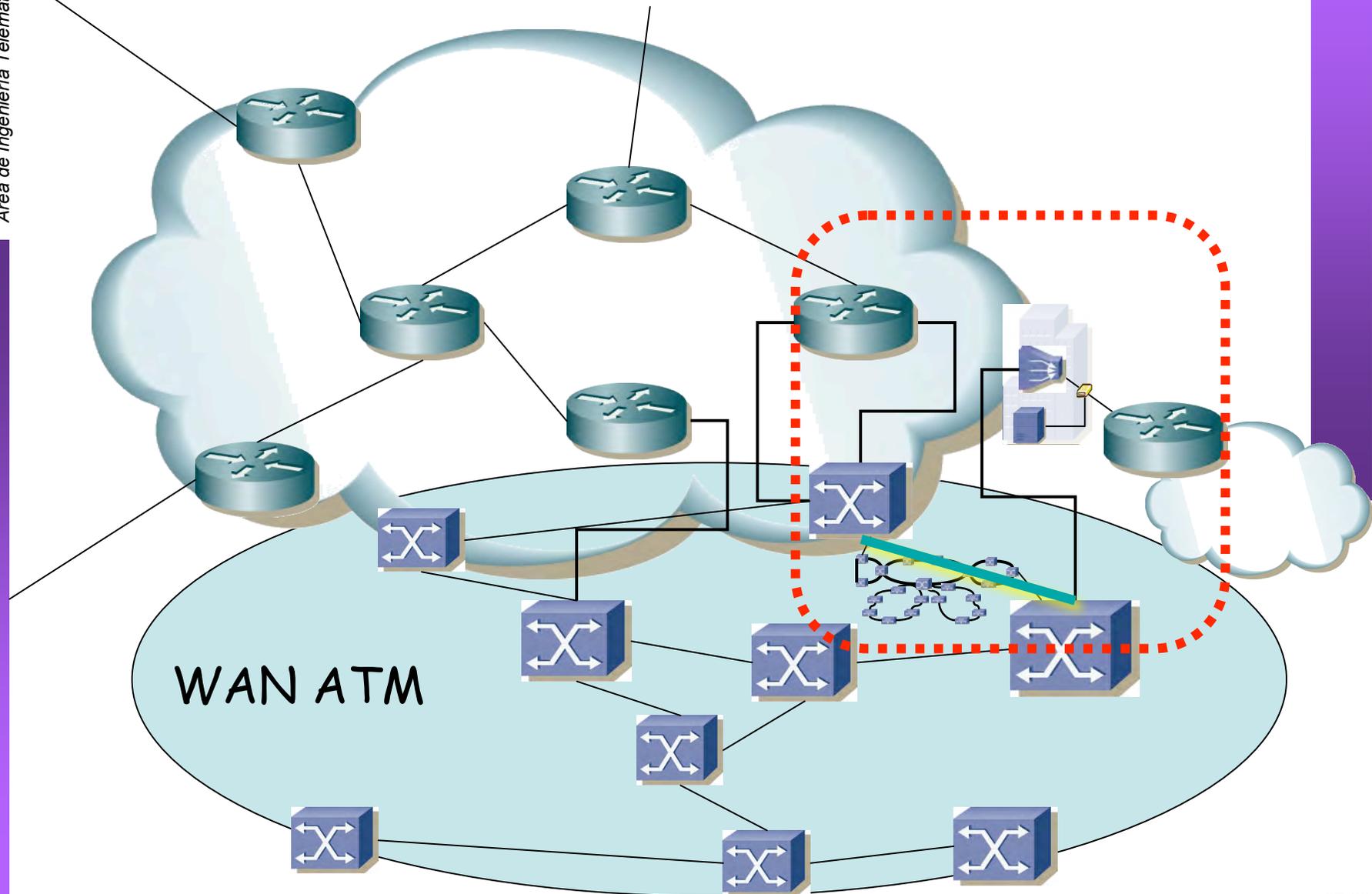
# SDH

- En 1 STM-1:
  - 1 señal de 140Mbps (E4) ó
  - 3 señales de 34/45 Mbps (E3/T3)
- Cada VC-3 puede sustituirse por 21 señales de 2Mbps (E1)



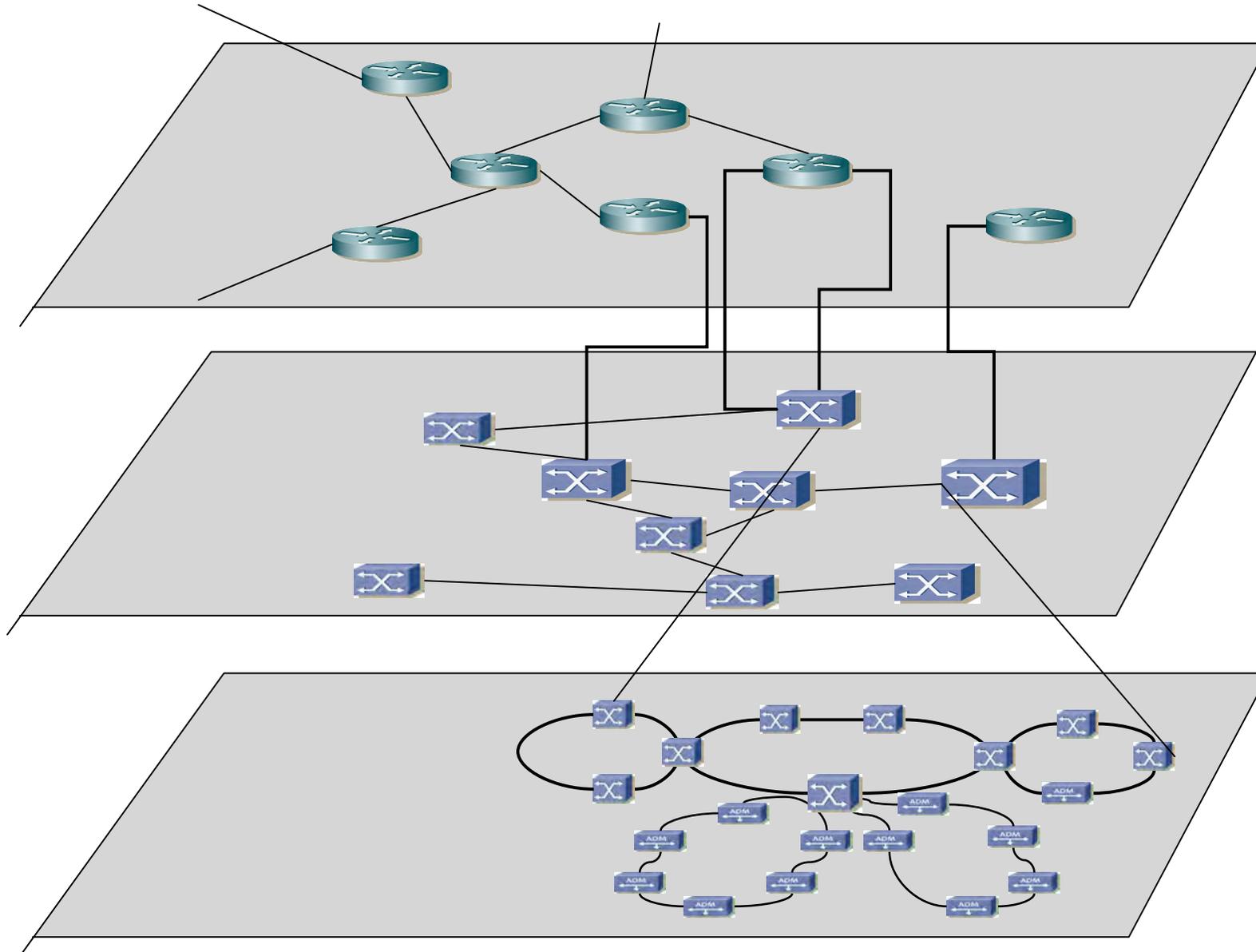


# Tecnologías WAN





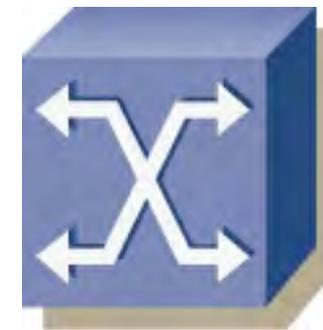
# Estructura multicapa





# ATM

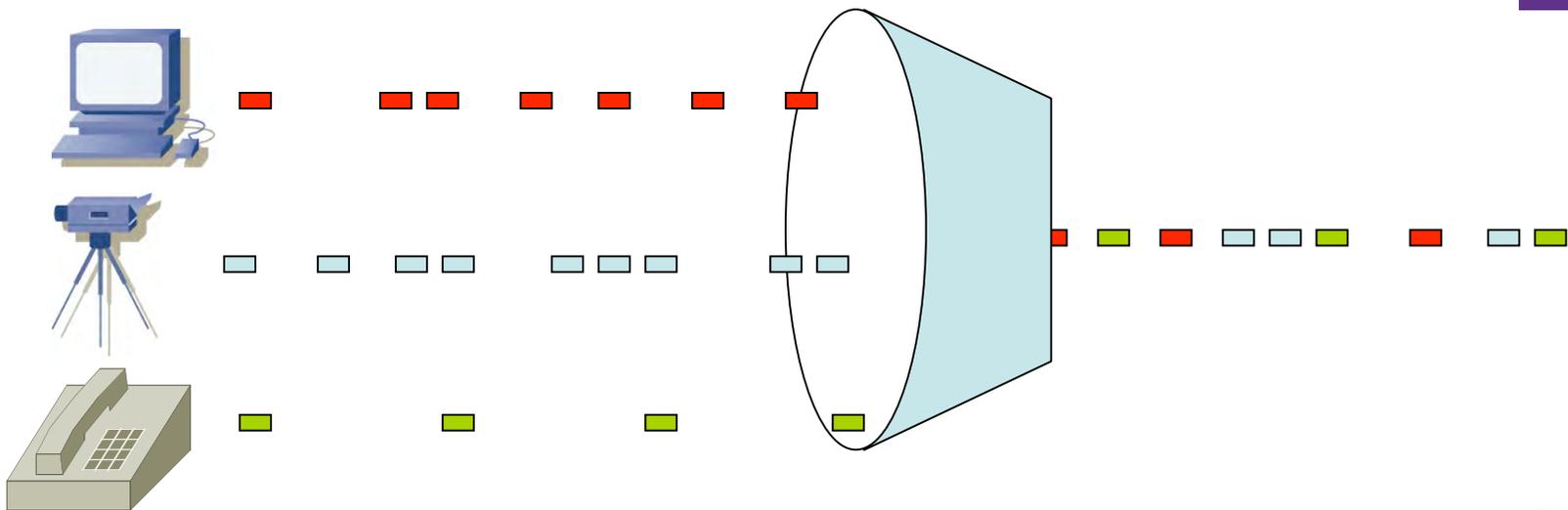
- ATM = *Asynchronous Transfer Mode*
- Seleccionada por la ITU como tecnología para la RDSI de banda ancha (BISDN)
- Orientado a conexión (circuitos virtuales): permite ofrecer capacidad garantizada y retardo acotado
- Una red para todo tipo de tráfico (voz, vídeo y datos)
- Conmutación de “celdas”: paquetes pequeños de tamaño constante
- No asegura que lleguen
- Mantiene el orden de las celdas





# Introducción

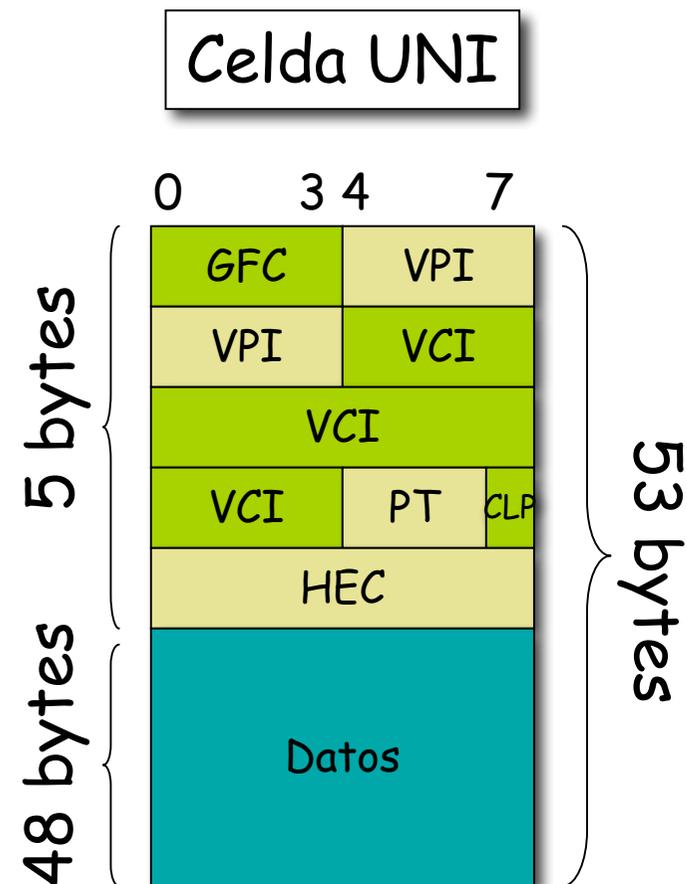
- ATM = *Asynchronous Transfer Mode*
- Orientado a conexión (circuitos virtuales)
- Una red para todo tipo de tráfico (voz, vídeo y datos)
- Conmutación de “celdas”: paquetes pequeños de tamaño constante
- No asegura que lleguen pero mantiene el orden de las celdas
- Mínima funcionalidad en la cabecera de las celdas
- Aprovecha la *multiplexación estadística*





# Estructura básica de las celdas

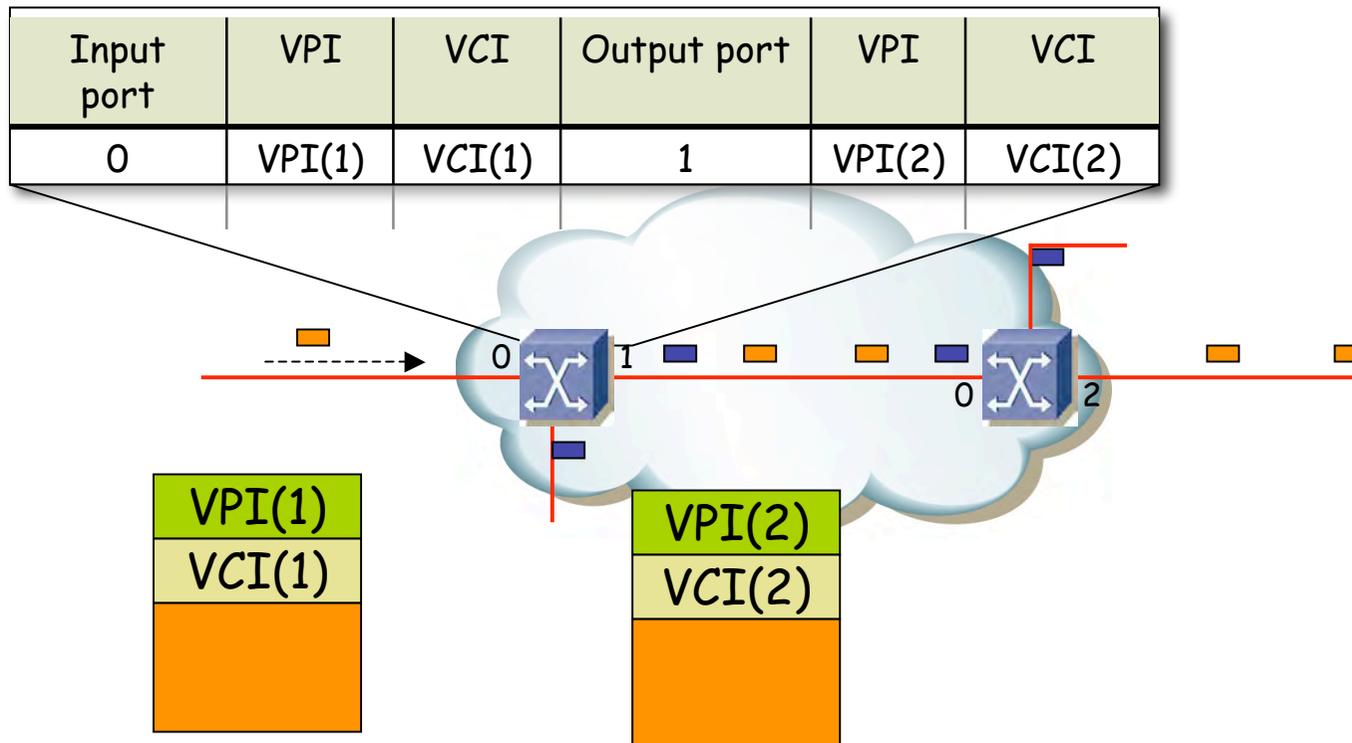
- 5 bytes cabecera  
+ 48 bytes datos  
= 53 bytes
- Simplifica el conmutador
- **VPI** = *Virtual Path Identifier*
- **VCI** = *Virtual Circuit Identifier*





# Cómo funciona ATM

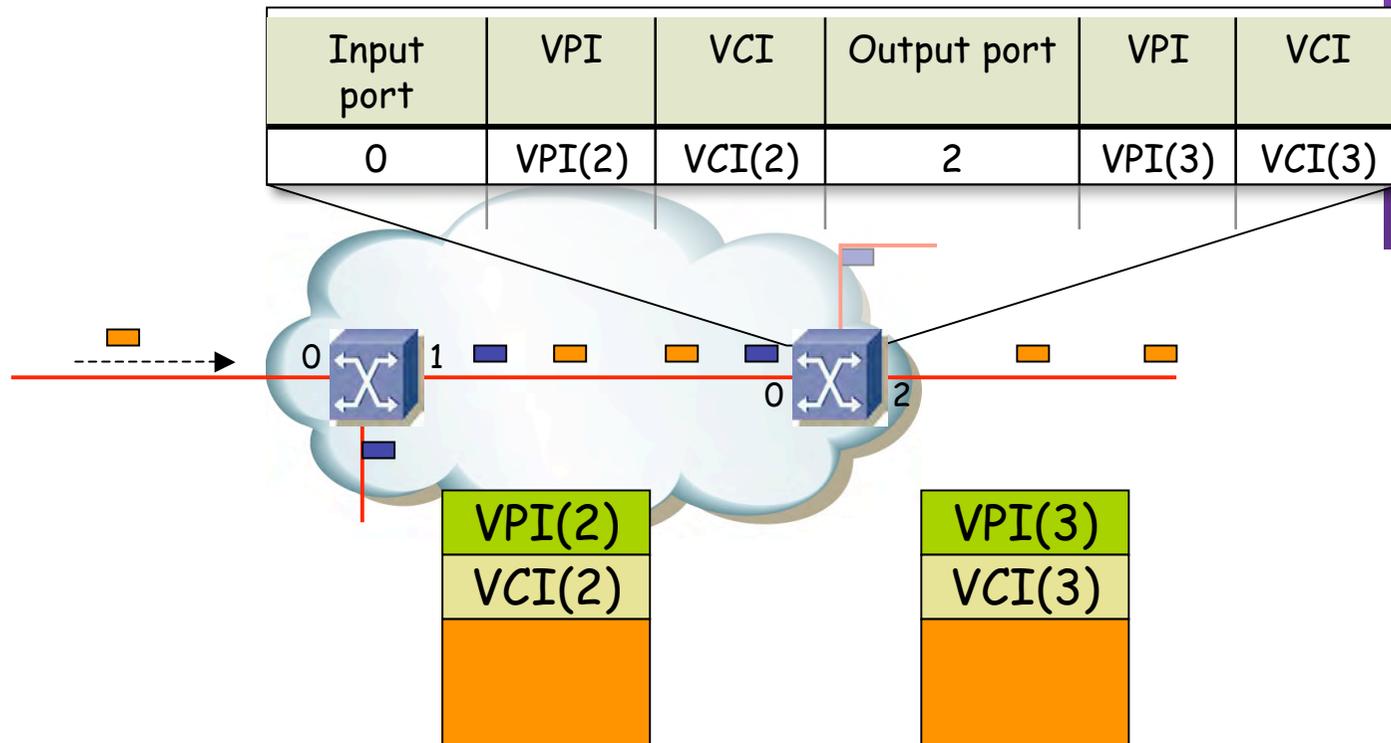
- Orientado a conexión
- Circuitos virtuales
- VPI/VCI identifica al circuito
- Solo tiene sentido localmente al nodo
- Circuitos full-duplex
- Se establecen mediante gestión o señalización





# Cómo funciona ATM

- Orientado a conexión
- Circuitos virtuales
- VPI/VCI identifica al circuito
- Solo tiene sentido localmente al nodo
- Circuitos full-duplex
- Se establecen mediante gestión o señalización





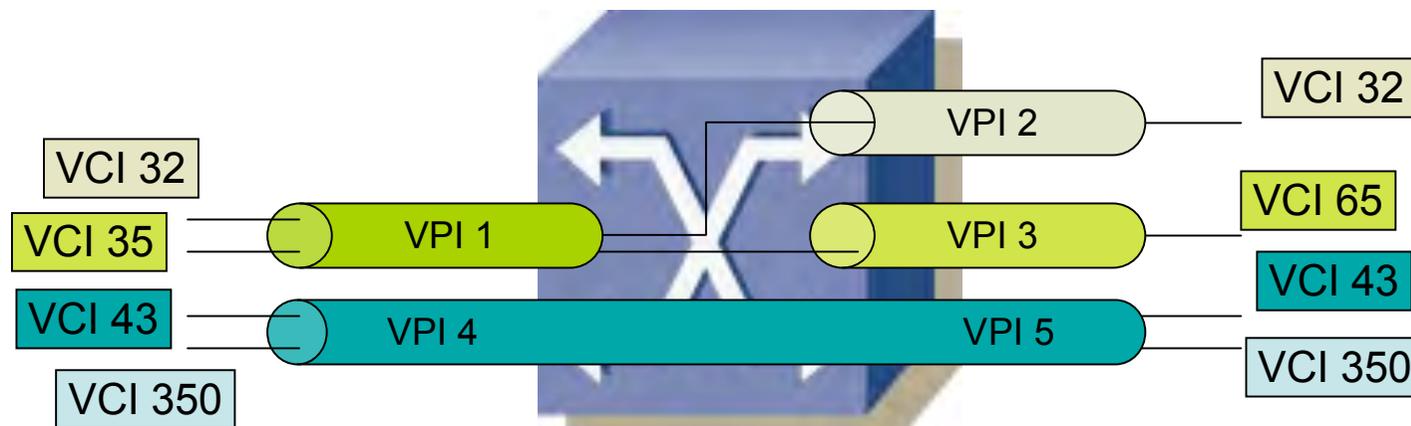
# Conexiones en los conmutadores

## VCC: Virtual Channel Connection

- La conmutación depende tanto del VPI como del VCI

## VPC: Virtual Path Connection

- La conmutación depende solo del VPI
- Usadas en el backbone



Input port	VPI	VCI	Output port	VPI	VCI
0	1	32	1	2	32
0	1	35	1	3	65
0	4	X	1	5	X



# Calidad de servicio

- ATM ofrece varias clases de QoS
- Para servicios de tiempo real:
  - CBR: *Constant Bit Rate*
  - rt-VBR: *real-time Variable Bit Rate*
- Para servicios no RT:
  - nrt-VBR: *non-real-time Variable Bit Rate*
  - ABR: *Available Bit Rate*
  - UBR: *Unspecified Bit Rate*
  - GFR: *Guaranteed Frame Rate* (solo para VCCs)



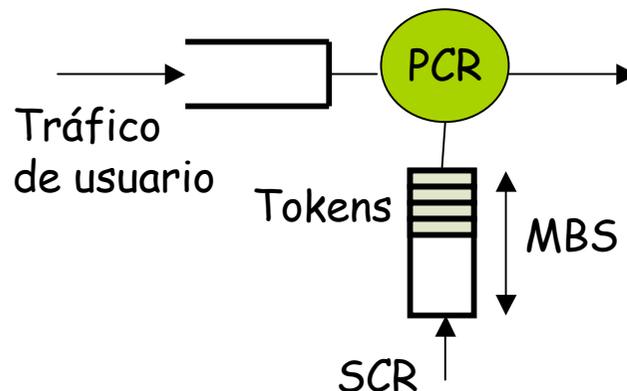
# Parámetros y descriptores de tráfico

- Los parámetros de tráfico describen las características del tráfico generado por una fuente
  - PCR (*Peak Cell Rate*)
  - SCR (*Sustainable Cell Rate*)
  - MBS (*Maximum Burst Size*)
  - MCR (*Minimum Cell Rate*)
  - MFS (*Maximum Frame Size*)
- El descriptor de tráfico de una fuente es el conjunto de parámetros de tráfico
- El descriptor de tráfico de una conexión incluye:
  - El descriptor de tráfico de la fuente
  - El CDVT (*Cell Delay Variation Tolerance*)
  - La técnica para decir qué celdas cumplen con los requisitos



# CAC, *Policing* y *Shaping*

- *Connection Admission Control*
  - Durante el establecimiento de la conexión
  - Acciones para determinar si se permite o no
- *Policing*
  - Acciones sobre las celdas que exceden el contrato de tráfico
  - Las marca con CLP=1
  - Serán la primeras en descartarse en caso de congestión
- *Traffic Shaping*
  - Obligar a que se cumpla los parámetros de tráfico





# UPC (*Usage Parameter Control*)

- Acciones que lleva a cabo la red para:
  - Verificar que el tráfico del usuario cumple con lo acordado
  - Forzar a que lo cumpla
- GCRA (*Generic Cell Rate Algorithm*)
  - Para cada celda dice si cumple con el contrato
  - La UPC puede usar GCRA u otro equivalente
  - Es un *Leaky-Bucket* de estado continuo



# CBR

- Conexiones que requieren una cantidad de BW continuo y estático
- Parámetros: PCR
- Máxima prioridad
- QoS síncrona garantizada
- Usos:
  - Voz de tasa constante
  - Vídeo
  - Datos
  - Emulación de circuitos TDM

# rt-VBR

- Parámetros: PCR, SCR, MBS
- Usos:
  - Aplicaciones con requisitos de retardo y variación del mismo
  - Vídeo y audio comprimido



# nrt-VBR

- Parámetros: PCR, SCR, MBS
- No asegura límites en el retardo

# UBR

- No hay garantías
- Se puede especificar un PCR para CAC y UPC
- Usos
  - Datos



# ABR

- Soporta que cambien las características de límite de tráfico ofrecidas por la red
- Emplea control de tráfico mediante realimentación para ajustar la tasa de la fuente
- RM-cells (*Resource Management*)
- No acota el retardo o la variación del mismo
- Parámetros:
  - PCR
  - MCR (*Minimum Cell Rate*)

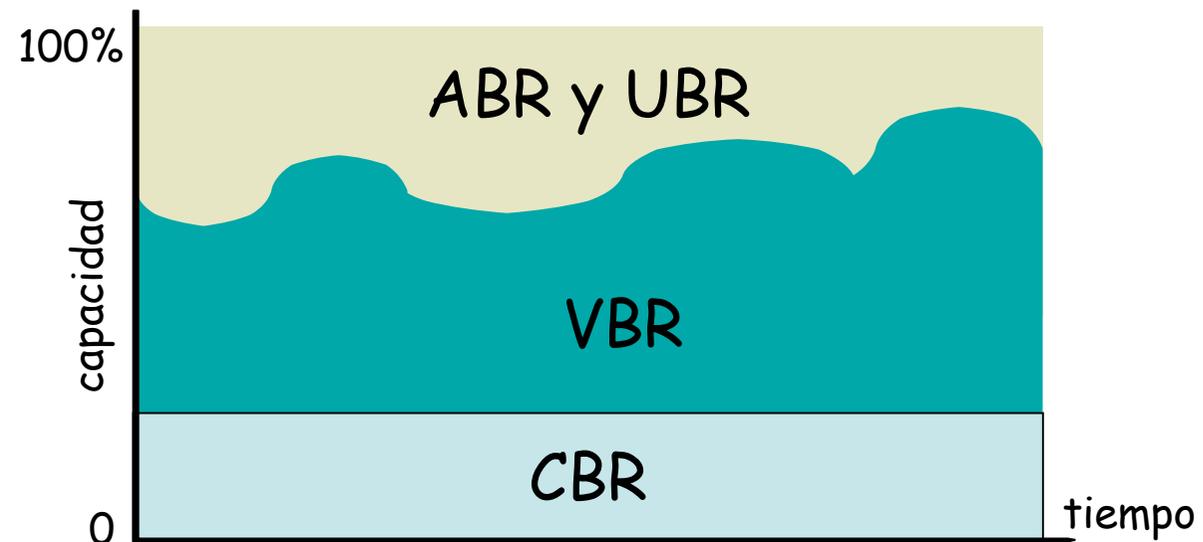


# GFR

- Para tramas AAL-5
- La red intenta descartar tramas en vez de celdas
- Parámetros:
  - PCR, MCR, MBS
  - MFS (*Maximum Frame Size*)

# UBR+ (UBR-G)

- UBR + MCR





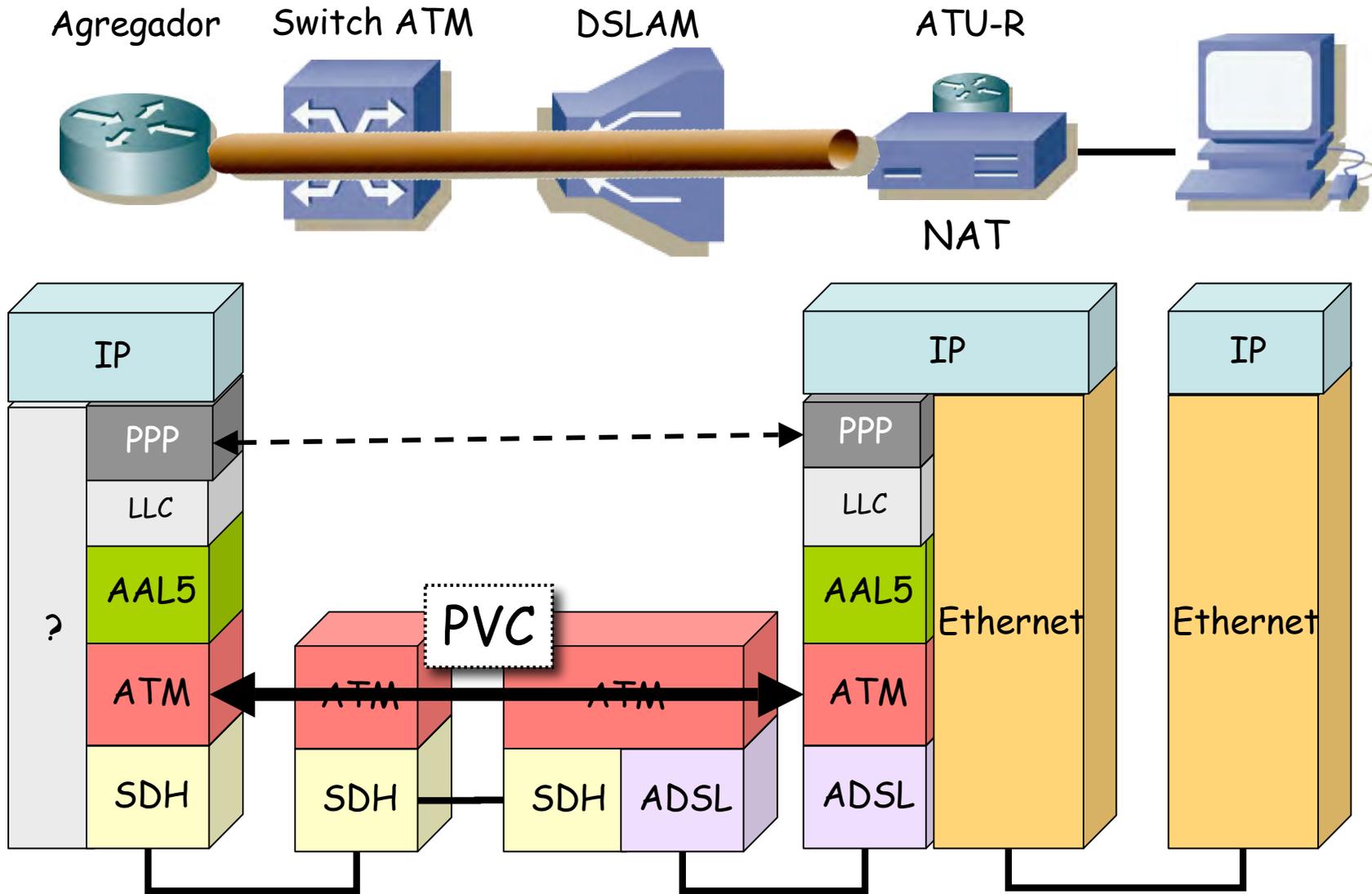
# ATM

- Tecnología muy importante en troncales de ISPs
- Importante conocerla pues ofrece QoS
- Pero su uso no se limita a la troncal...





# Ejemplo: PPPoA





# Internet: ¿Posibles evoluciones?

## **Sin cambios**

- Emplear solo técnicas en el nivel de aplicación
- Sobredimensionar BW

## ***Integrated Services (IntServ) :***

- Cambios fundamentales en Internet para soportar reservas de BW extremo a extremo
- Requiere nuevo y complejo software en host y routers

## ***Differentiated services (DiffServ) :***

- Menos cambios en la infraestructura
- Permite diferenciar la calidad de servicio entre tipos de tráfico
- No da garantías firmes de retardo o BW





# DiffServ

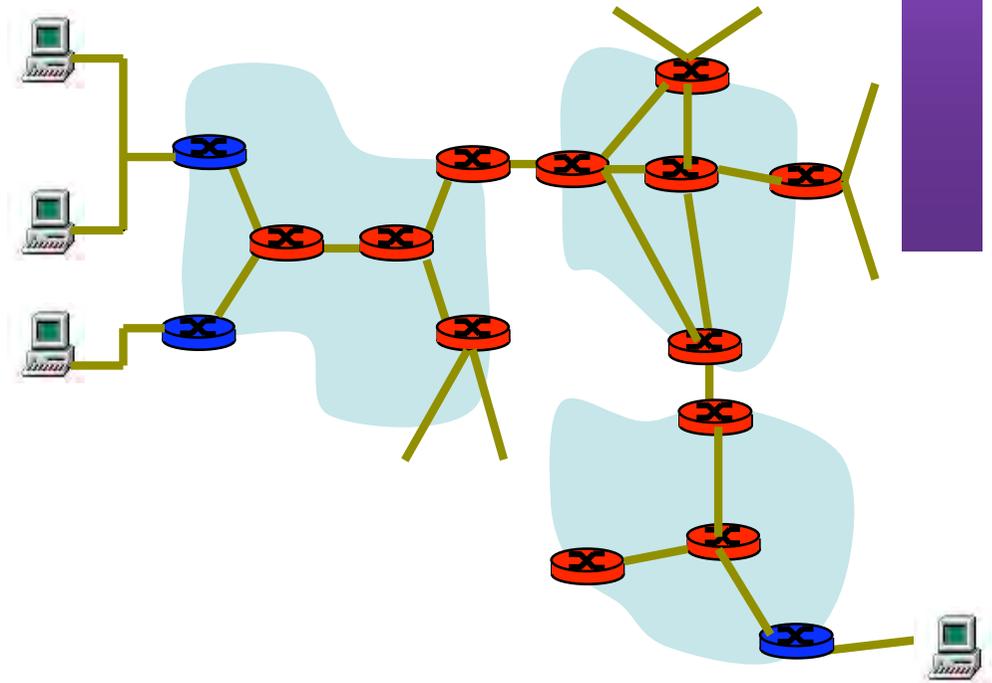
## Problemas de *IntServ*:

- Escalabilidad y sobrecarga: mantiene información de cada flujo en cada router

## *DiffServ*:

- Diferencia clases de tráfico (etiquetado)
- Aplica diferente planificación según la clase
- Simple de aplicar en el núcleo
- Clasificación (etiquetado) complejo en la frontera

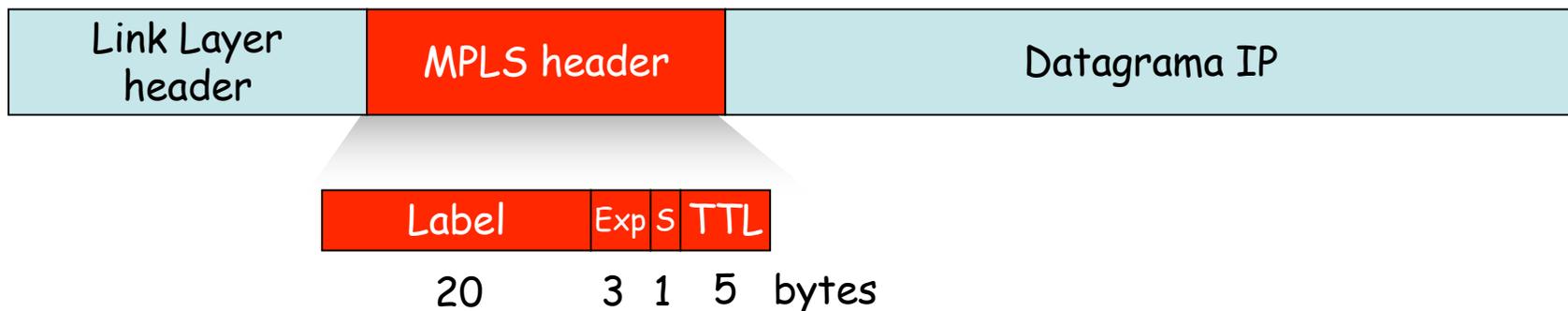
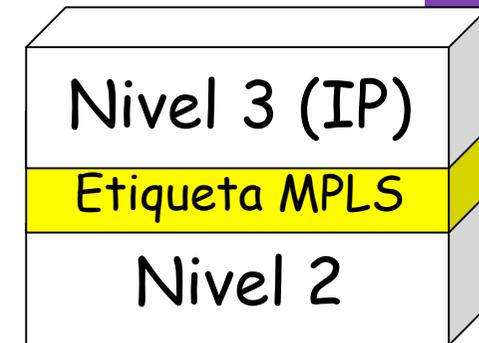
- Router frontera 
  - Etiqueta el tráfico
- Router del núcleo (core) 
  - *Buffering* y *scheduling* según las etiquetas





# MPLS

- Combinación de circuitos virtuales (ATM) e IP
- Añade una etiqueta a los paquetes IP
- Se encamina en función de esa etiqueta
  - Mayor rapidez para tomar las decisiones
  - Requiere establecer antes el camino (*LSP=Label Switched Path*)
- Se dice que es de nivel 2.5
- Puede emplear diferentes tecnologías de nivel 2
- Señalización para establecer los caminos (RSVP-TE)
- Permite hacer *Traffic Engineering*



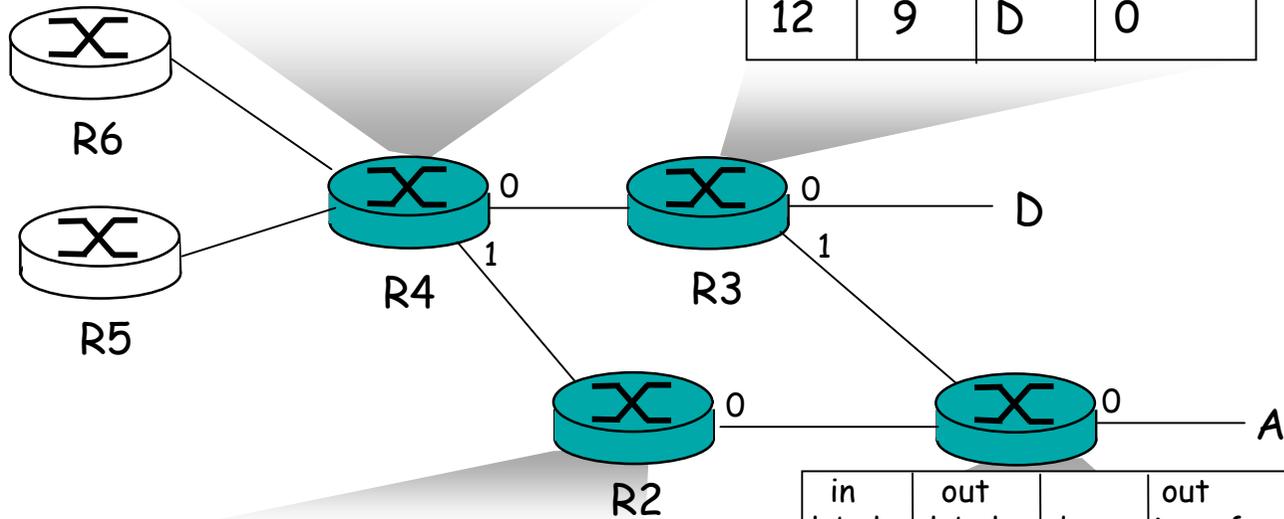


# MPLS forwarding tables

LSR (Label Switched Router) : 

in label	out label	dest	out interface
	10	A	0
	12	D	0
	8	A	1

in label	out label	dest	out interface
10	6	A	1
12	9	D	0



in label	out label	dest	out interface
8	6	A	0

in label	out label	dest	out interface
6	-	A	0



**SERVICIOS EN LA WEB Y DISTRIBUCIÓN DE CONTENIDOS**  
*Área de Ingeniería Telemática*

# Modelado de tráfico



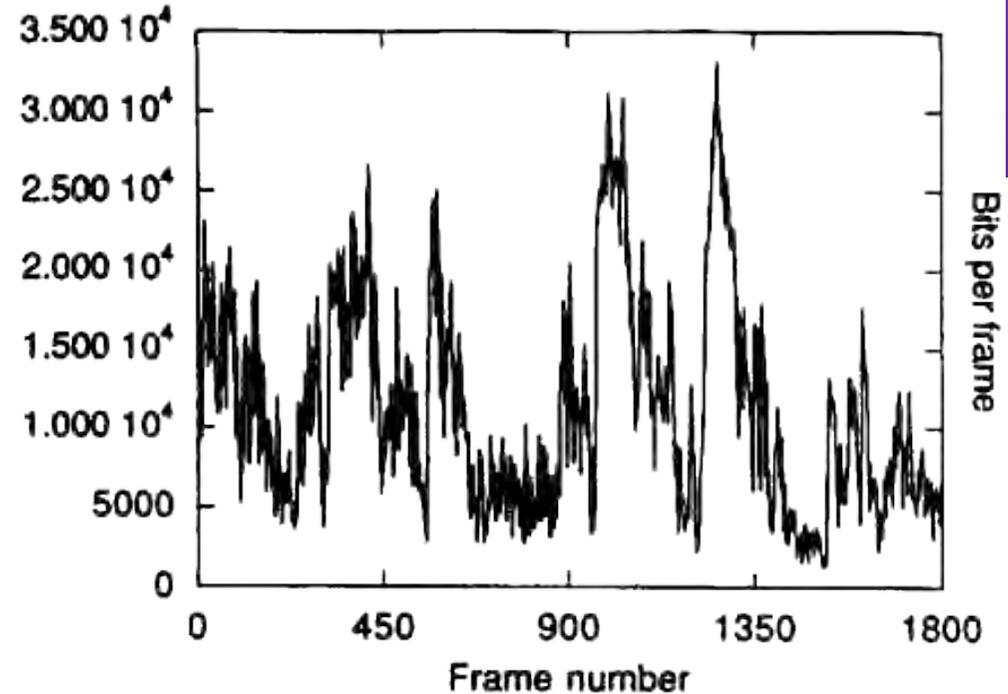
# Tráfico generado por fuentes de vídeo

## Interesa modelar:

- PDF de tamaños de tramas
- Autocorrelación

## Para:

- Predecir pérdidas en buffers
- Dimensionar BW y buffer necesario





# Papers que vamos a ver

- D.P. Heyman and T.V. Lakshman, “*Source Models for VBR Broadcast-Video Traffic*”, IEEE/ACM Transactions on Networking, vol. 4, n° 1, Feb 1996
- W. Liao and V. O.K. Li, “*The Split and Merge Protocol for Interactive Video-on-Demand*”, IEEE Multimedia, vol. 4, n° 4, Oct-Dec 1997
- M.H. Kabir, R.G. Manning and G.C. Shoja, “*A CBR-Streaming Scheme for VBR-encoded Videos*”, Proceedings of IEEE Pacific Rim Conference on Communications, Computers and signal Processing 2005
- E. Veloso, V. Almeida, W. Meira, A. Bestavros and S. Jin, “*A Hierarchical Characterization of a Live Streaming Media Workload*”, IEEE/ACM Transactions on Networking, vol. 14, n° 1, Feb 2006



**SERVICIOS EN LA WEB Y DISTRIBUCIÓN DE CONTENIDOS**  
*Área de Ingeniería Telemática*

# *Streaming*

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

Programa de Tecnologías para la gestión distribuida de la  
información