

# QoS: Arquitecturas y elementos

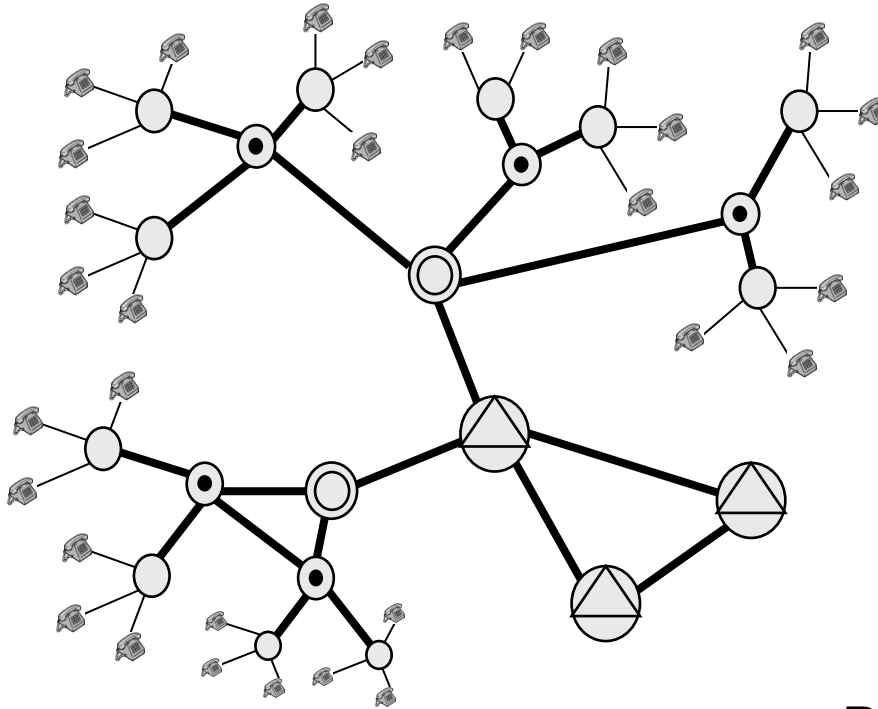
Area de Ingeniería Telemática  
<http://www.tlm.unavarra.es>

Grado en Ingeniería en Tecnologías de  
Telecomunicación, 3º

# Objetivos

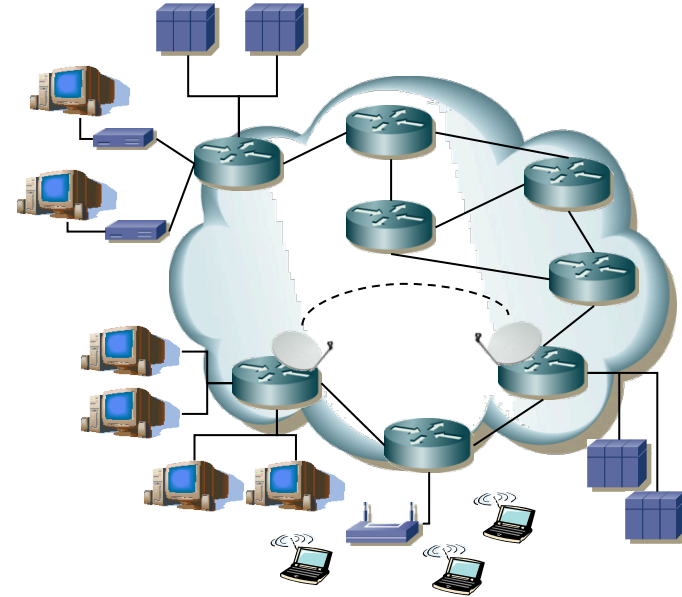
- Conocer los elementos básicos en cualquier arquitectura que ofrezca QoS

# Escenarios históricos



## PSTN

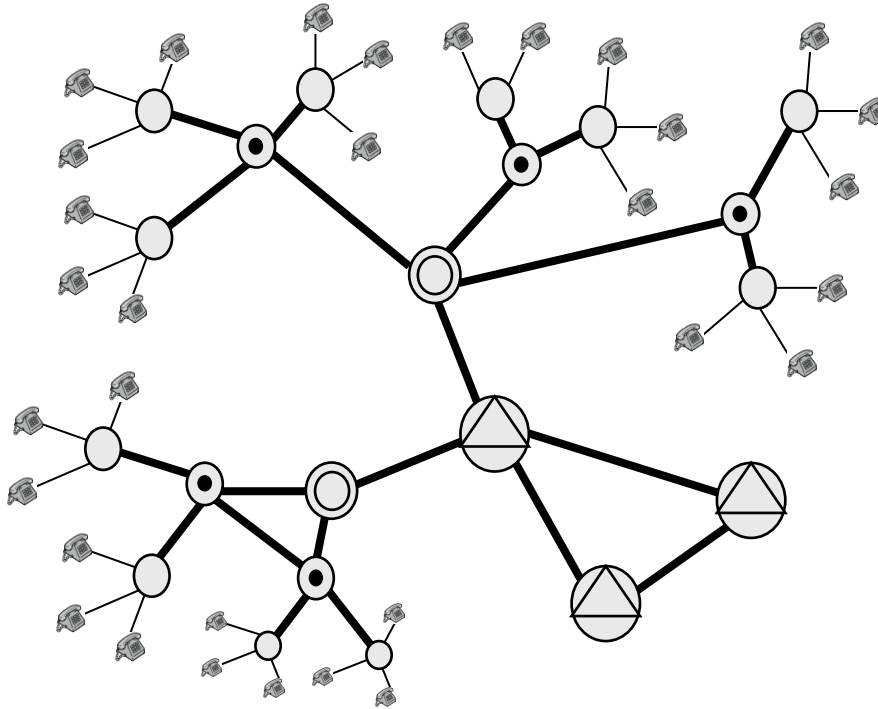
- Conmutación de Circuitos
- BW fijo y garantizado
- Retardo fijo y acotado
- Diseñada para tráfico de voz
- Para datos BW sin usar



## Redes de Conmutación de paquetes

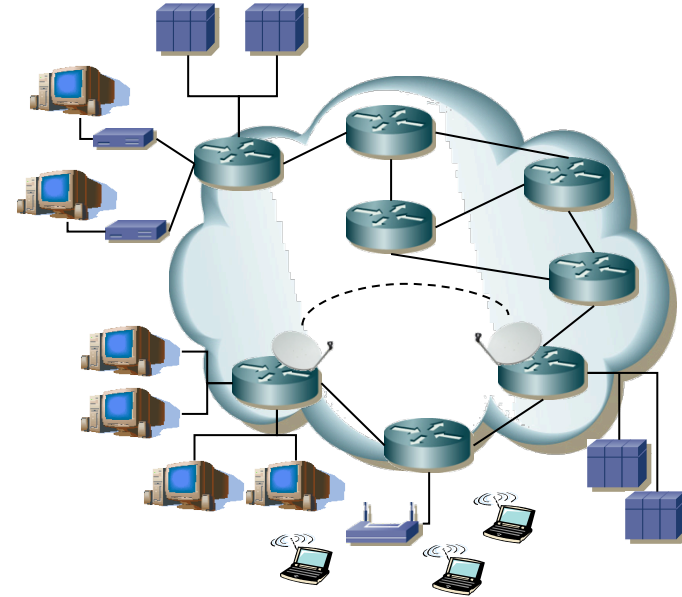
- *No single points of failure*
- Circuitos virtuales (ATM, FR, X.25)
  - *RT traffic*
  - Voz sobre circuitos no era comercial hasta hace poco (no económico)
- ATM busca las características de la PSTN

# Escenarios históricos



## PSTN

- Conmutación de Circuitos
- BW fijo y garantizado
- Retardo fijo y acotado
- Diseñada para tráfico de voz
- Para datos BW sin usar

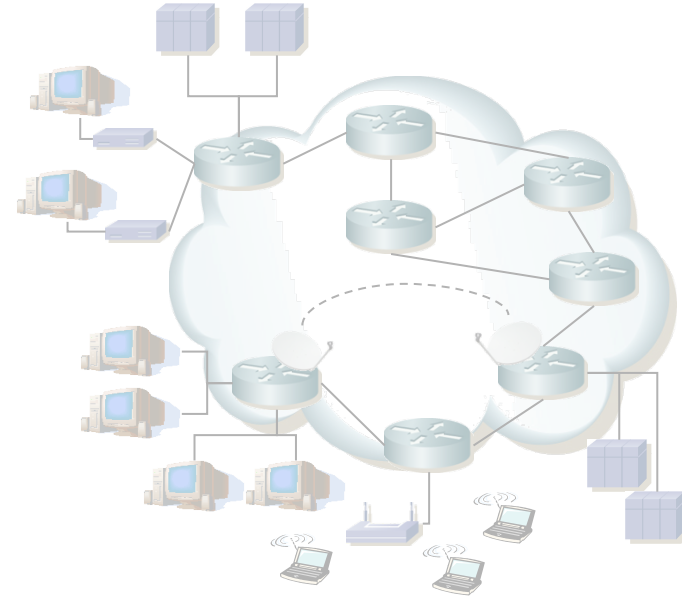
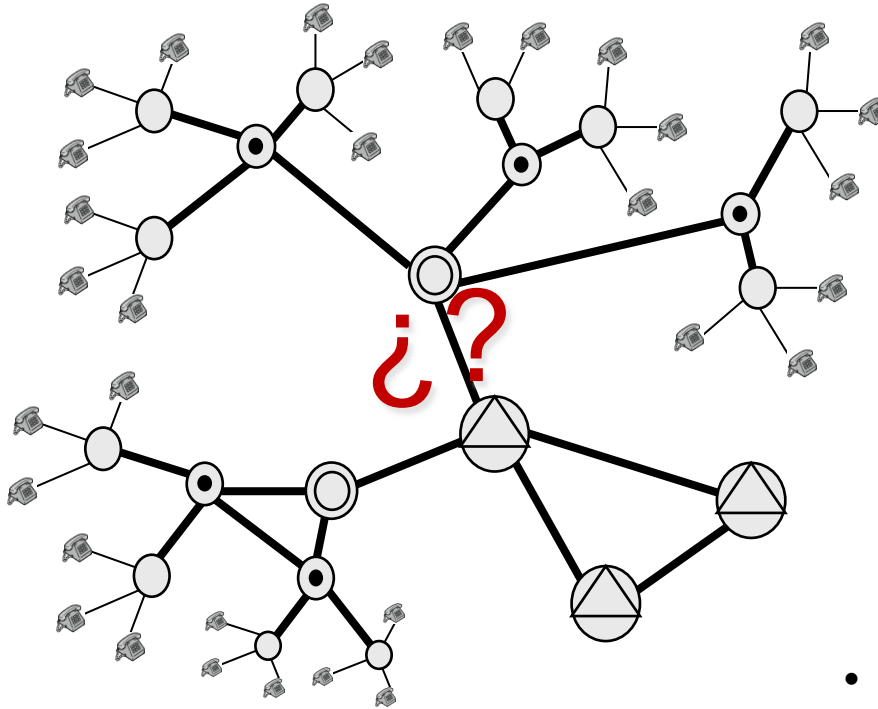


## Tecnología IP

- Conmutación de paquetes
- *Best Effort*
- TOS, no usado realmente

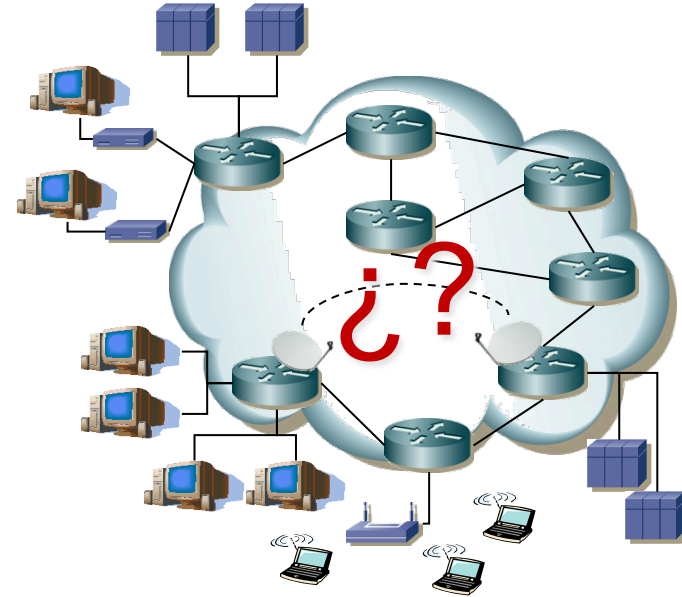
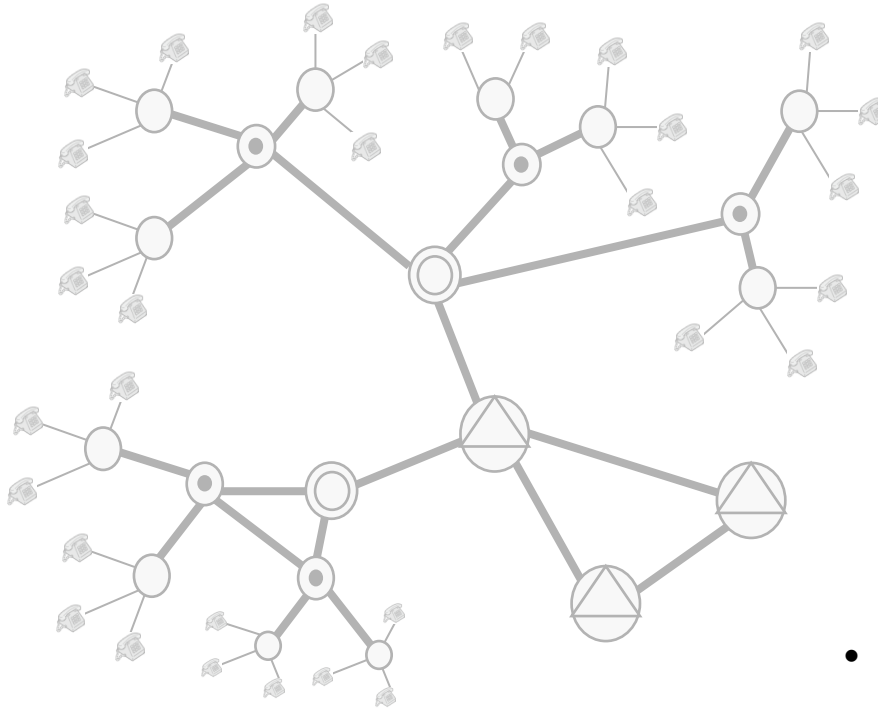
Versión	Header Length	TOS	Longitud		
16-bit identifier			D	M	13-bit fragmentation offset
TTL	Protocolo	Header checksum			
Dirección IP origen					
Dirección IP destino					
[opciones]					
[Datos]					

# Escenarios históricos



- ¿ Tener dos infraestructuras ?
  - Más caro
  - Más equipos
  - Más BW sin usar
  - Gestión independiente
- ¿Por qué no usar simplemente la PSTN?
  - No optimizada para datos
  - Arquitectura rígida
  - Ineficiente en asignaciones de BW
  - Inadecuada para sesiones cortas, de tasa variable, multipunto, etc

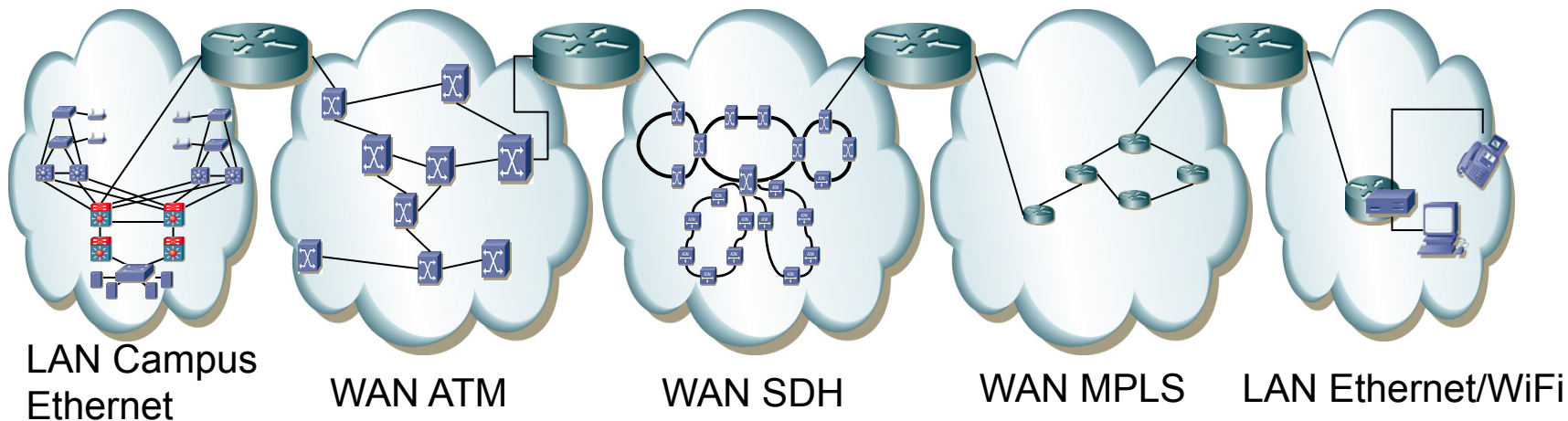
# Escenarios históricos



- ¿ Usar una red de paquetes ?
  - Deberá tener soporte de QoS
  - Servicio equivalente para la voz
  - Aprovechar multiplexación estadística
  - ATM seleccionada para RDSI de banda ancha
  - No extendida
- ¿IP?
  - Best Effort aunque sea sobre tecnologías con QoS
  - Separar los flujos IP en flujos de tecnología con QoS
  - o añadir QoS a IP
  - Para ello tratar de forma diferenciada al tráfico de datos
  - Hoy solo en el dominio de redes concretas

# IP QoS

- Tecnologías de capa 2 ofrecen QoS (ATM, Ethernet...)
- IP es la tecnología de nivel de red extremo a extremo más extendida
- Diferentes tecnologías capa 2 pueden ser empleadas en el camino capa 3
- Lo más razonable es mapear QoS de capa 3 en la QoS de capa 2 de cada tecnología en cada salto
- En vez de mapear capacidades de capa 2 de un salto en el siguiente



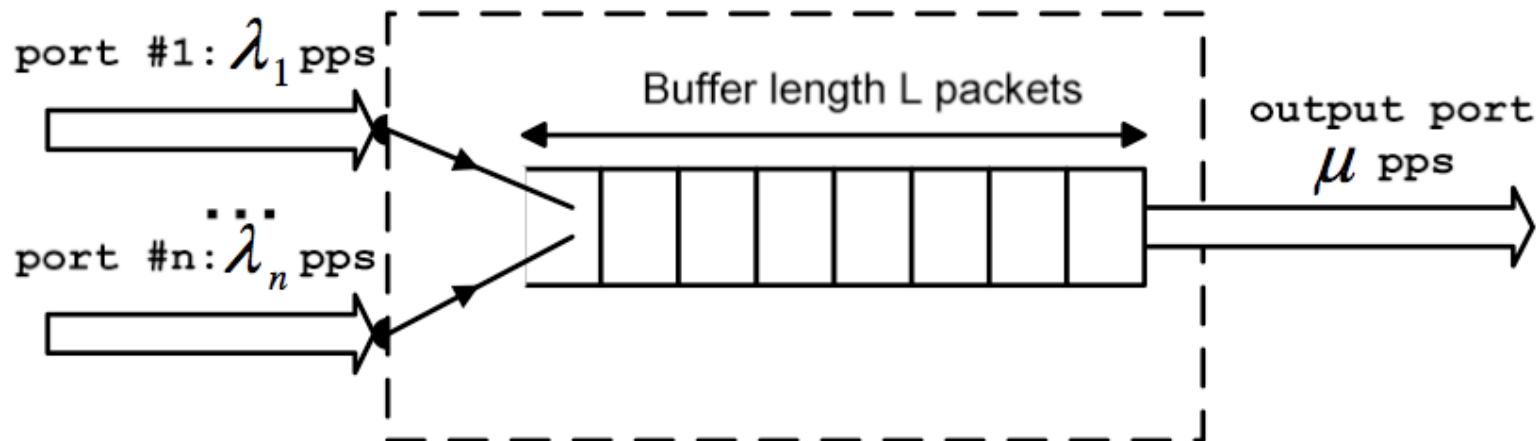
# Servicios / Arquitecturas

- Best Effort Service
- Integrated Services
- Differentiated Services



# Servicio *Best Effort* (BE)

- Se trata igual a todo el tráfico
  - Sin separación entre flujos
  - Sin diferenciación entre paquetes
- No garantiza ningún SLA
- Ante congestión
  - Crecen los retardos sin control
  - Pérdidas sin control



# IntServ

- IETF
- Para cada flujo (puede ser agregado) reserva recursos en todo el camino
- Orientado a conexión
- Cada router del trayecto ha de tomar nota y efectuar la reserva solicitada (guardan estado)
- Requiere un protocolo de señalización que soporten todos los routers: RSVP
- No requiere modificar los protocolos existentes
- RSVP no hace la reserva, solo la señala
- Poco utilizado salvo en algunos escenarios de videoconferencia
- Poco escalable
- Resurgiendo con MPLS

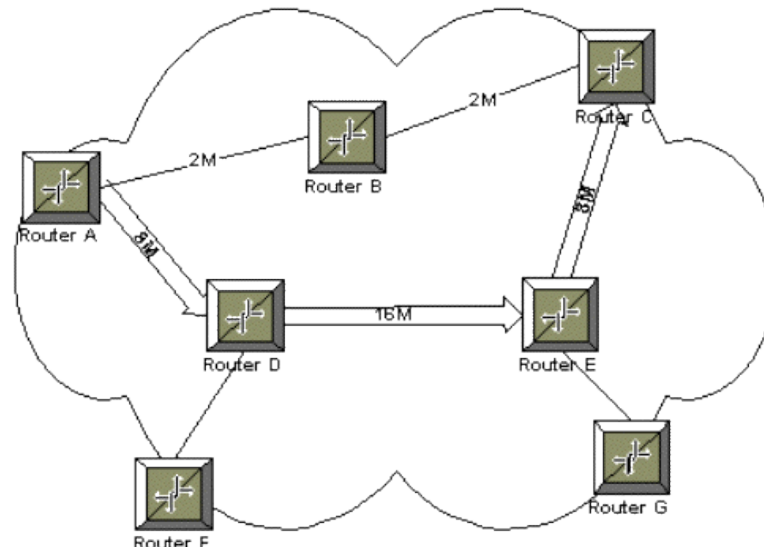
# DiffServ

- IntServ no escala bien
- RFC 2475, 2638
- Clasificar el tráfico en pocas clases
- Clasifican los *ingress routers* (complejidad en la frontera) con un *codepoint* en la cabecera IP
- DiffServ mapea en cada nodo el *codepoint* en el paquete a un PHB en concreto
- PHB = *Per Hop Behavior*
  - El tratamiento que se le da al paquete en cuestión de scheduling y gestión de cola en ese nodo
  - El mapeo *codepoint* ↔ *PHB* debe ser configurable
- No es sensible a los requisitos de un flujo individual

# Elementos

- **QoS routing / Traffic Engineering**
  - Encontrar caminos “buenos” para flujos con requisitos específicos de QoS
  - Usar la red de forma eficiente: aumentar la probabilidad de aceptar peticiones futuras
  - Es complicado:
    - Información precisa sobre el estado de la red es difícil de mantener
    - Calcular caminos que cumplan requisitos de QoS es costoso (computacionalmente hablando)
  - *Constraint-based Routing*
    - Calcular caminos teniendo en cuenta no solo QoS sino también políticas

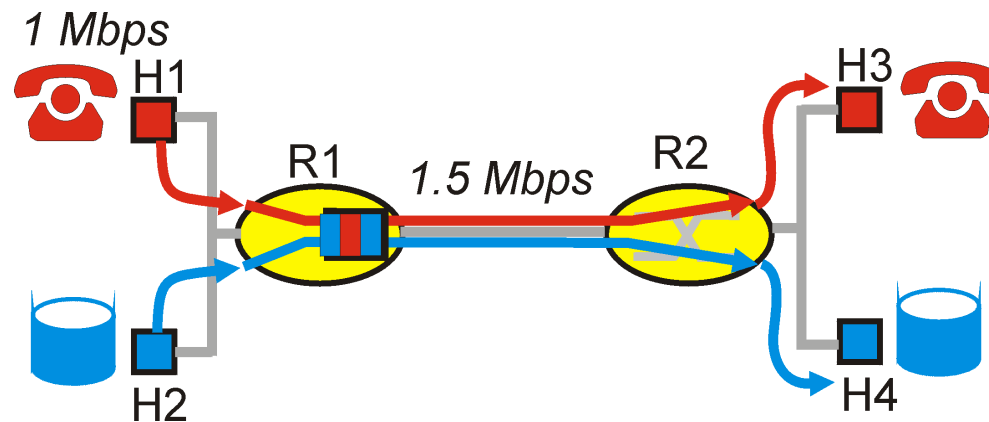
A → C: 4M



# Elementos

- **Clasificación / Marcado**

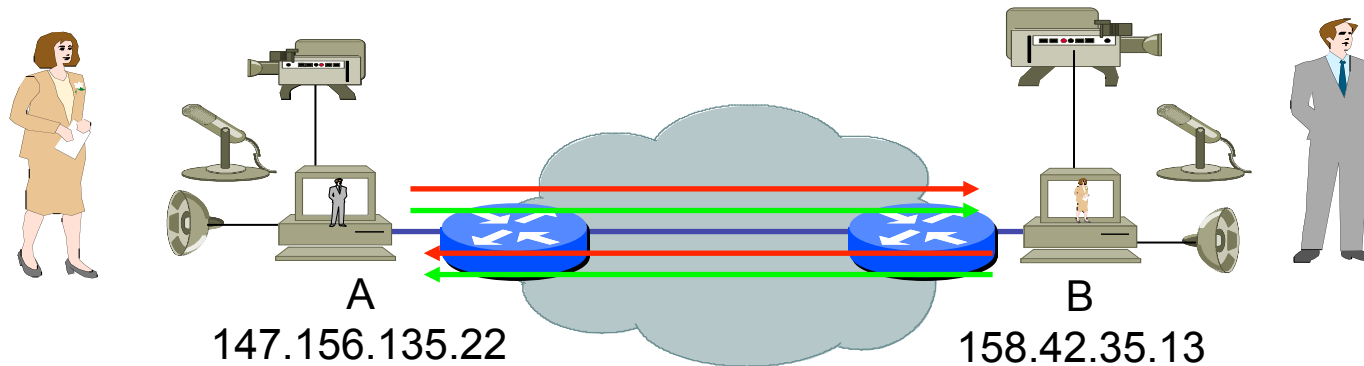
- ¿Cómo distinguir entre flujos?
- Ejemplo: Teléfono IP a 1Mbps, comparte enlace de 1.5Mbps con FTP
  - Ráfagas de FTP pueden congestionar el enlace y causar fallos en el audio
  - Queremos dar prioridad al audio sobre el FTP



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

# Concepto de flujo en QoS

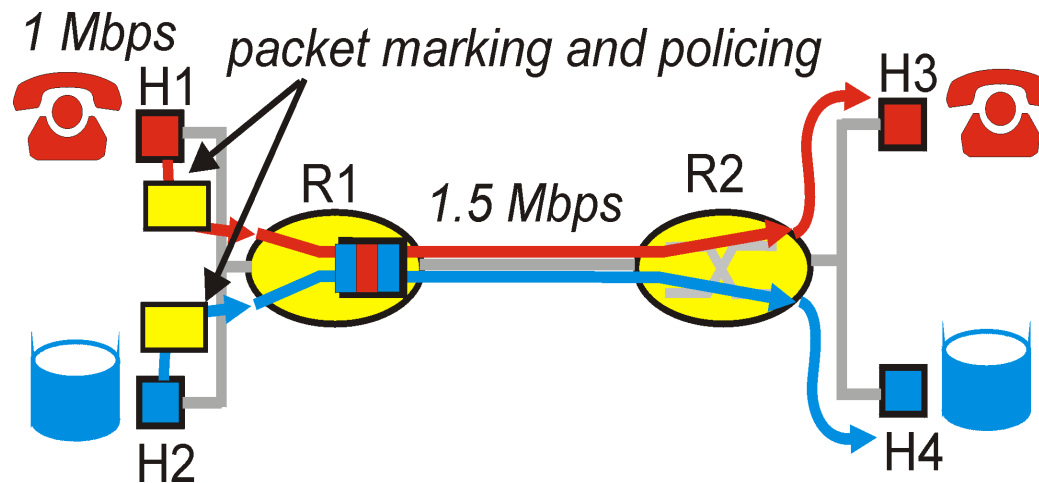
- Secuencia de datagramas que se produce como resultado de una acción del usuario y requiere la misma QoS
- Normalmente es simplex (unidireccional)
- Es la entidad más pequeña a la que los routers pueden aplicar una determinada QoS
- Ejemplo: una videoconferencia estaría formada por cuatro flujos, dos en cada sentido, uno para el audio y otro para el vídeo.
- Los flujos pueden agruparse en clases; todos los flujos dentro de una misma clase reciben la misma QoS.



- Flujo vídeo A->B: 147.156.135.22:2056 -> 158.42.35.13:4065
- Flujo audio A->B: 147.156.135.22:3567 -> 158.42.35.13:2843
- ← Flujo vídeo B->A: 158.42.35.13:1734 -> 147.156.135.22:6846
- ← Flujo vídeo B->A: 158.42.35.13:2492 -> 147.156.135.22:5387

# Elementos

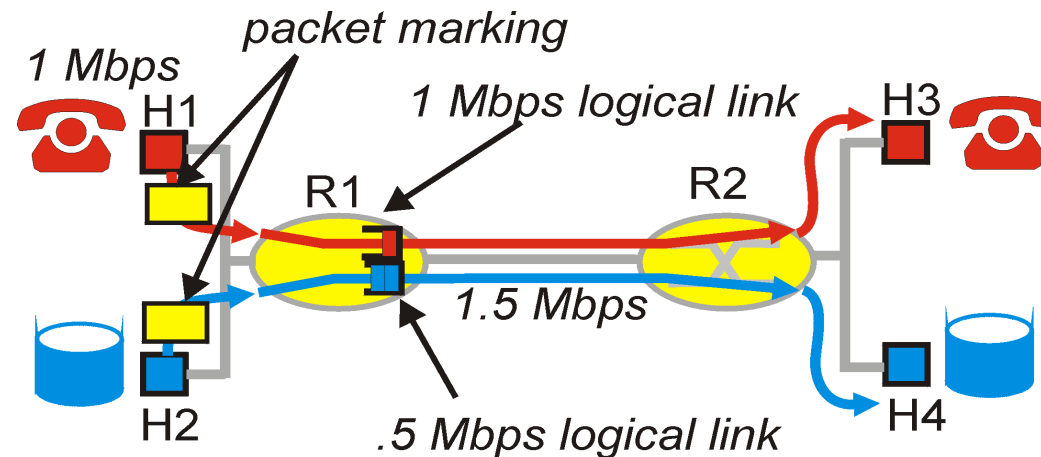
- **Traffic shaping y policing**
  - Marcar, descartar o retrasar el tráfico en exceso
  - ¿Qué sucede si las aplicaciones no se comportan como deben?
    - Por ejemplo la aplicación de audio envía más de lo previsto
    - Necesitamos forzar que las fuentes se comporten como se ha acordado



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

# Elementos

- **Gestión de cola**
  - ¿Qué paquetes tirar si se llena?
- **Planificación de recursos (*scheduling*)**
  - El recurso normalmente es el enlace
  - ¿Cómo organizar a los paquete que deben enviarse?
  - ¿Dar prioridades? ¿Repartir la capacidad?

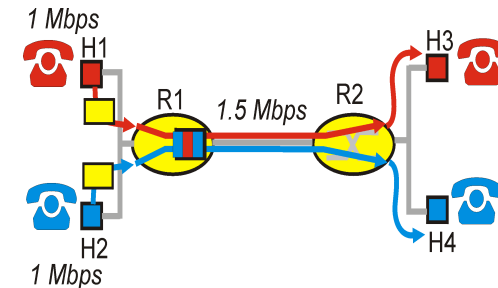
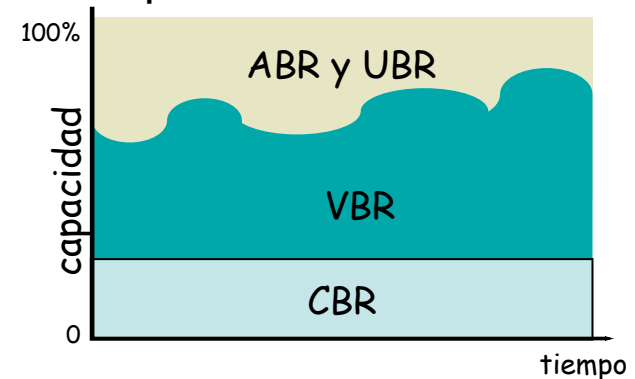


Mientras se ofrece aislamiento es deseable emplear los recursos de forma eficiente (*work conserving*): *scheduling* (en todos los routers del camino)



# Elementos

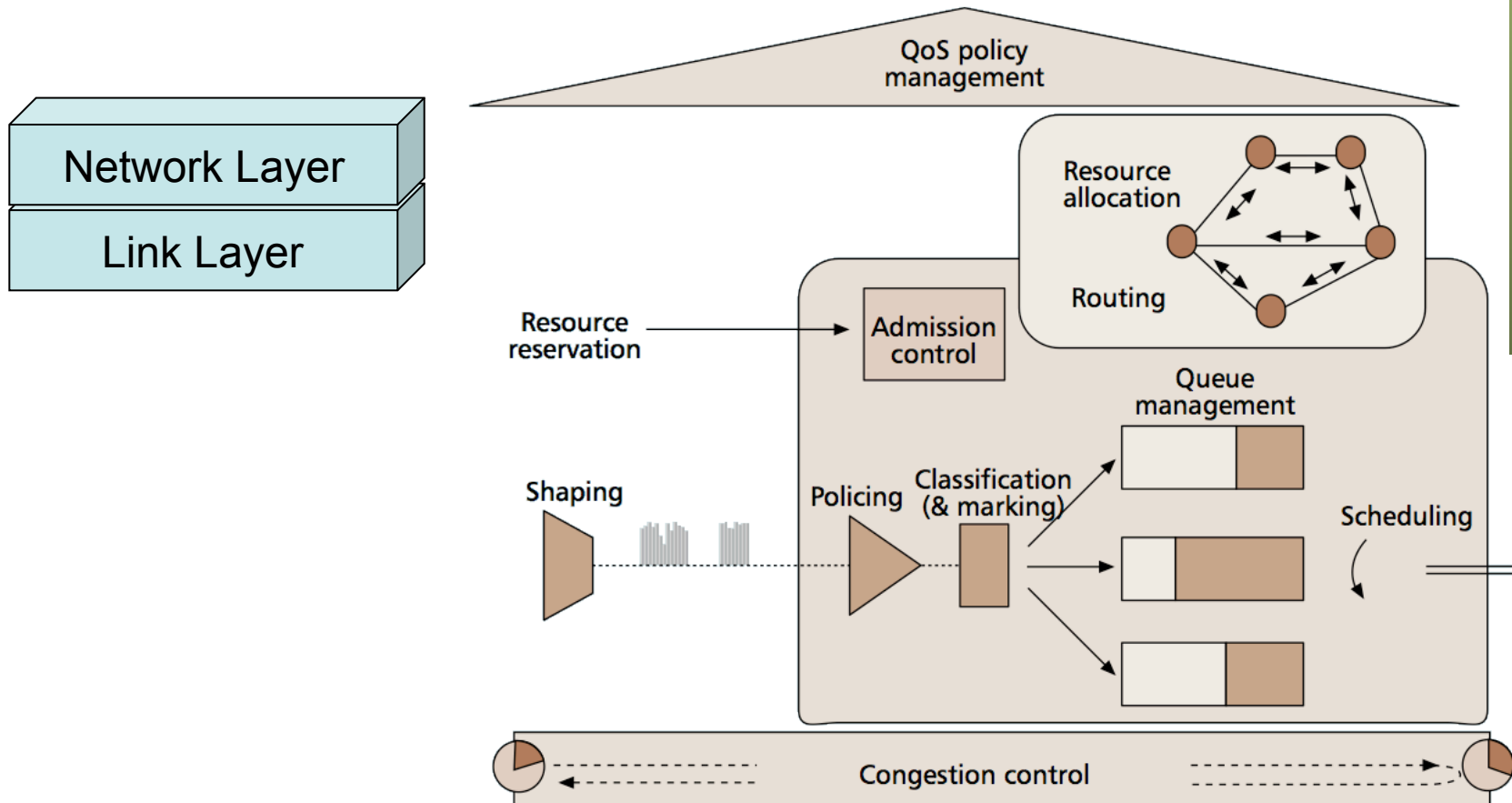
- **Connection Admission Control (CAC)**
  - ¿Puede la red cursar el nuevo flujo de tráfico manteniendo los parámetros de QoS ofrecidos a todos los usuarios?
  - Aceptarlo en la red o rechazarlo
  - No se pueden satisfacer las demandas más allá de la capacidad del enlace
  - Es algo básico desde siempre en redes de conmutación de circuitos porque hay reserva de recursos
  - Con flujos CBR el cálculo es relativamente simple
  - ¿Y con flujos VBR ?



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

# Localización de los elementos

- Router, switch o similar
- Tienen sentido en más de una capa



# Resumen

- *Best Effort / IntServ / DiffServ*
- *Connection Admission Control (CAC)*
  - ¿Puede la red cursar el nuevo flujo de tráfico manteniendo los parámetros de QoS ofrecidos a todos los usuarios?
- *Planificación de recursos (scheduling)*
  - El recurso normalmente es el enlace
  - ¿Cómo organizar a los paquete que deben enviarse?
  - ¿Dar prioridades? ¿Repartir la capacidad?
- *Traffic shaping y policing*
  - Marcar, descartar o retrasar el tráfico en exceso
- *Monitorización*
  - Analizar la cantidad de tráfico que entra en la red
- *QoS routing / Traffic Engineering*
- *Clasificación*
  - ¿Cómo distinguir entre flujos?
- *Gestión de cola*
  - ¿Qué paquetes tirar si se llena?

QoS for networked applications

