Tema 1: QoS
Quality of Service

Introducción

¿ Qué es esto de la calidad ?

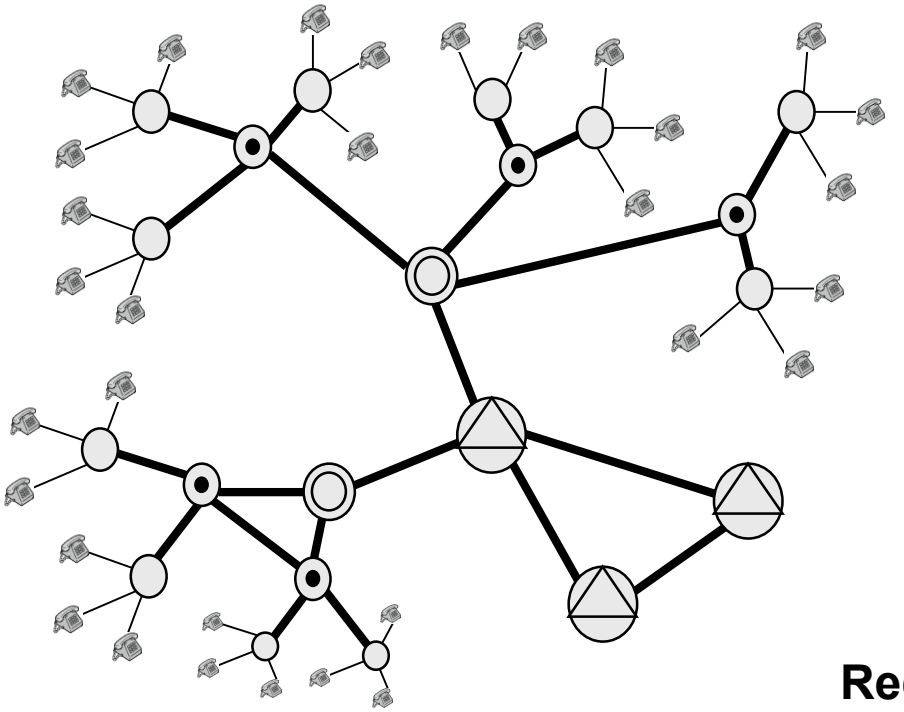
Para el usuario final

- Para un usuario experimentado es normal que una llamada con un ordenador tenga diferente calidad que una por teléfono fijo o que una por móvil
- ¡ Aunque todas se cursen por la misma red !
- Es simplemente aquello a lo que está acostumbrado
- Si nunca ha usado un móvil esperará una calidad similar a la PSTN y se quejará
- Lo mismo si nunca ha usado VoIP
- La calidad es relativa a las expectativas
- Lo mismo con el precio, si está acostumbrado a una tarifa plana o gratis le extrañará pagar

Para el técnico

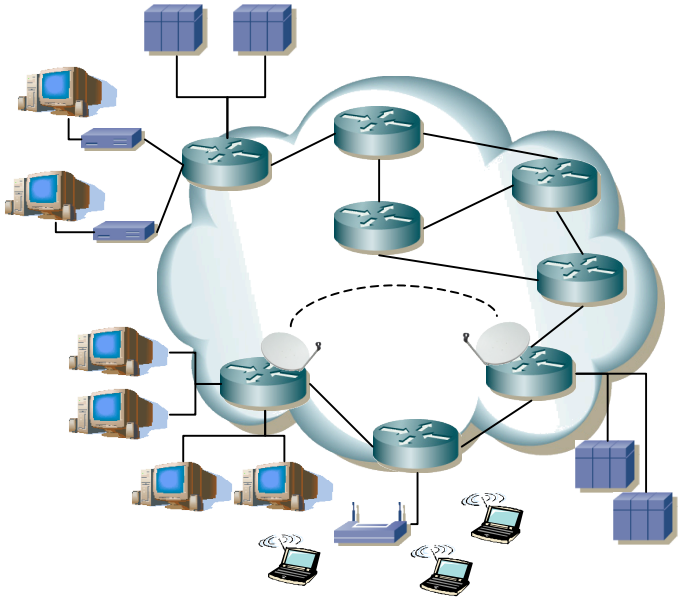
- En función de parámetros estadísticos:
  - Throughput, pérdidas, retardo, quejas de usuarios
  - Más absolutos y medibles
- Formalizados en SLAs

Escenarios históricos



PSTN

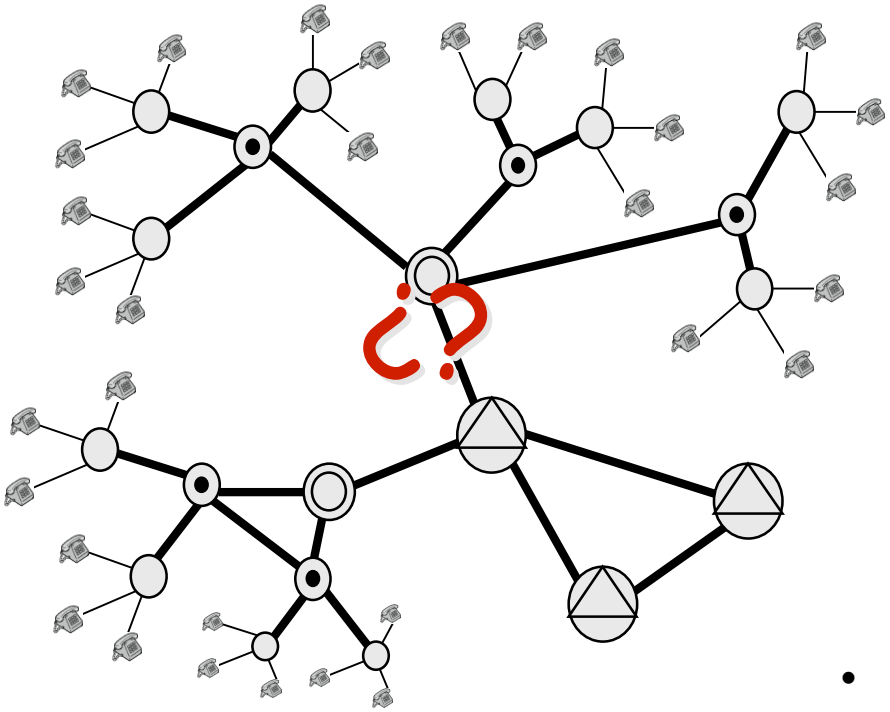
- Conmutación de Circuitos
- Fixed BW
- RT traffic (voz)

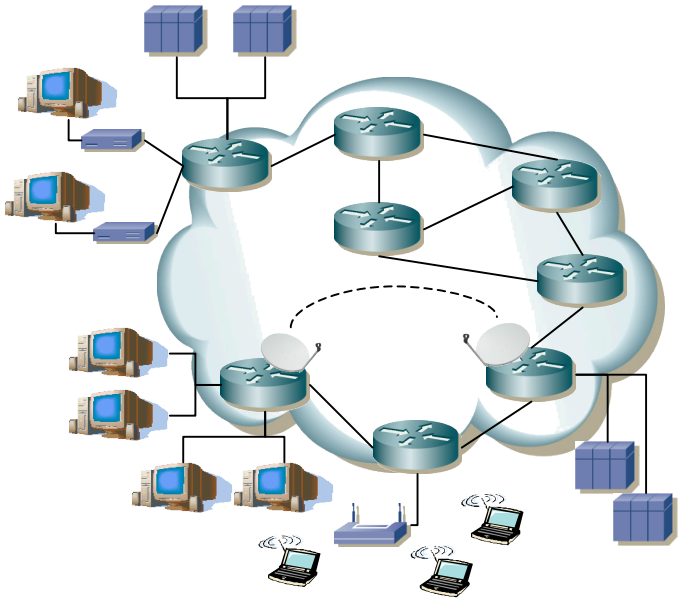


Red de Conmutación de paquetes

- 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





- ¿ 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
  - Desperdicio en asignaciones de BW
  - Inadecuada para sesiones cortas, de tasa variable, multipunto, etc

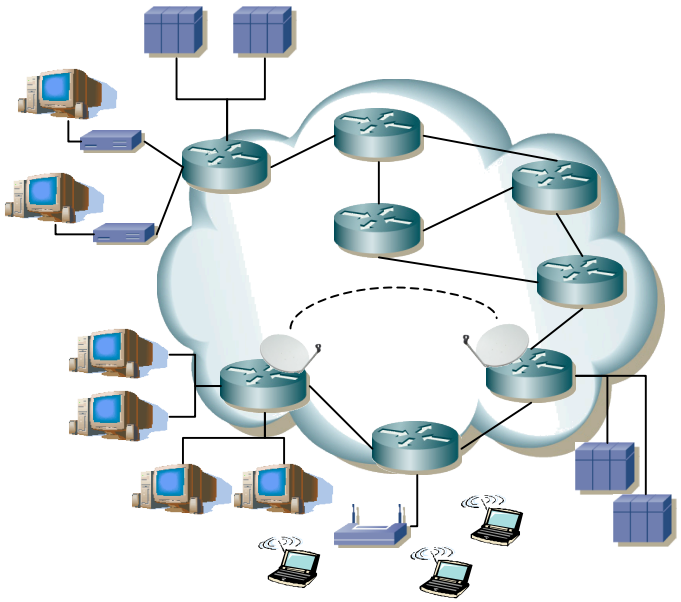
¿ ISDN ?

- *Integrated Services*
- Diseñada para voz, vídeo y datos
- Infraestructura de Conmutación de Circuitos
- Define cómo transportar datos en una arquitectura que soporta voz nativa
- Nunca llegó a implantarse de forma extensa (por ejemplo por culpa de las políticas de precios)

IP

- Conmutación de paquetes
- *Best Effort*
- Internet en conjunto sigue BE
- Converger de nuevo los 3 servicios
- Para ello incluirle QoS
- Tratar de forma diferenciada el tráfico de datos
- Estaba desde los comienzos con el TOS
- Hoy en día es una realidad pero solo en el dominio de redes concretas

IETF®



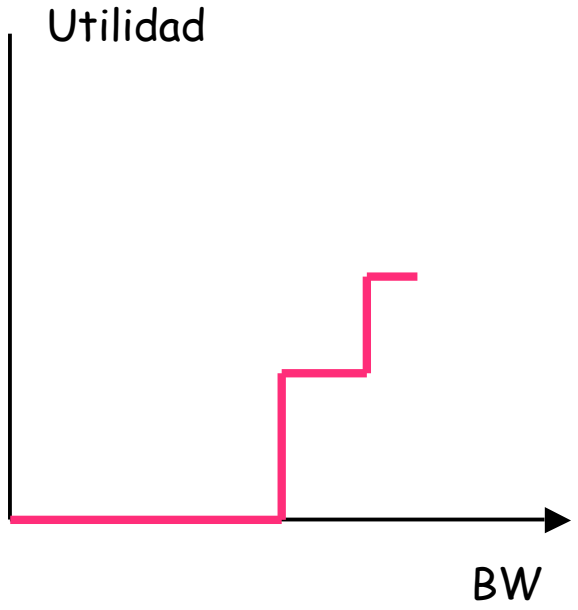
Versión	Header Length	TOS	Longitud			
16-bit identifier				DF	MF	13-bit fragmentation offset
TTL		Protocolo	Header checksum			
Dirección IP origen						
Dirección IP destino						
[opciones]						
[Datos]						

Service Level Agreements

- SLAs, dentro varios SLSs (*Service Level Specifications*)
- Acuerdo entre proveedor de servicio (la red) y el suscriptor (el cliente)
- Especifica la calidad de servicio que garantizará el proveedor
- La red mantendrá su promesa mientras los flujos de usuario se mantengan dentro de su especificación de tráfico
- Especifica las medidas que se tomarán si se incumple
- Gran cantidad de parámetros posibles según el servicio

Utilidad

- Aplicaciones son sensibles a:
  - Retardos
  - Pérdidas
- Por debajo de un umbral puede no ser útil el tráfico
- Ofrecer garantías de prestaciones para
  - Que el usuario esté satisfecho
  - Que los recursos se usen de forma óptima



¿ Quién necesita QoS ?

- Voz (IP telephony, radio?)
- Vídeo (streaming, videoconferencia)
- Ciertas aplicaciones de datos (generalmente elásticas)
  - *Transactional Data/Interactive Data* (SAP, Oracle…)
  - *Bulk Data* (backups, replicación en redes de contenidos…)
  - *Locally Defined Mission-Critical Data* (mayor que *transactional*)
- Resto:
  - *Best Effort*
  - Dejar BW para él
  - Gran cantidad de aplicaciones en una empresa (centenares)
  - Probablemente no se puedan clasificar todas, ¡no ahogarlas!
- ¿Queda algo?: *Scavenger Service*
  - *Less than BE*
  - Tráfico no deseado: DoS, Worms,
  - Web surfing a destinos no relacionados con el objetivo de la empresa
  - Si no se descarta se cursa solo en la capacidad que sobra

Requisitos de QoS de las aplicaciones

Aplicación	Fiabilidad	Retardo	Jitter	Ancho de Banda
Correo electrónico	Alta (*)	Alto	Alto	Bajo
Transferencia de ficheros	Alta (*)	Alto	Alto	Medio
Acceso Web	Alta (*)	Medio	Alto	Medio
Login remoto	Alta (*)	Medio	Medio	Bajo
Audio bajo demanda	Media	Alto	Medio	Medio
Vídeo bajo demanda	Media	Alto	Medio	Alto
Telefonía	Media	Bajo	Bajo	Bajo
Vídeoconferencia	Media	Bajo	Bajo	Alto

(*) La fiabilidad alta en estas aplicaciones se consigue automáticamente al utilizar el protocolo de transporte TCP

Ejemplo: Afecta a la calidad de voz

Paquetes perdidos

- Causan cortes y saltos
- Un paquete suele contener en torno a 20ms de muestras de voz
  - Si contiene menos, mayor ratio cabeceras/datos
  - Si contiene más, mayor retardo de formación
- Pérdida de 1 paquete se puede intentar recuperar (interpolación, etc)
- Pérdida de más de 1 paquete crea un corte que se nota claramente

Paquetes retrasados y jitter

- Si un paquete llega demasiado tarde es equivalente a una pérdida
- Si el retardo general end-to-end es demasiado grande se pierde interactividad
- El jitter (variación en el retardo) se puede recuperar con buffer en el receptor
- Si el jitter es demasiado grande el buffer ha de ser tan grande que de nuevo implica demasiado retardo

Causas de pérdidas

***Bit Errors* en los enlaces**

- Asumimos que están dimensionados para errores despreciables frente a la congestión

Congestión

- Buffer overflows en conmutadores/routers
- Descarte para alivio preventivo de la congestión
- Retardo excesivo

Ejemplos: Equipos

Switch D-Link DES-3028

Quality of Service (QoS)	+ Quality of Service (QoS)	+ 802.1p Priority Queues: 4 Queues
	+ Packet Classification Based on: - Switch Port - TCP/UDP Port Number - IPv4 - TOS - User Defined Packet Content - VLAN ID - MAC Address - 802.1p Priority - DSCP - Protocol Type	+ Support WRR/ Strict Mode

Juniper Routers Serie J

Traffic Management

Marking, policing & shaping
Class based queuing with prioritization
WRED
Queuing based on VLAN/DLCI/Interface/Bundles/Filters

MPLS

Layer 2 VPN
Layer 3 VPN
Label Distribution Protocol (LDP)
Resource Reservation Protocol (RSVP)
Circuit Cross-Connect (CCC)
Translation Cross Connect (TCC)

Voice Transport

FRF.12
Link Fragmentation and Interleaving (LFI)
Compressed Real-Time Protocol (CRTP)

ZyXEL ES-3124

Traffic Management and QoS

- Rate Limiting: Rule-based/Port-based bandwidth control, 64kbps granularity
- Supports Two-Rate-Three-Color (CIR/PIR)
- Port-based egress traffic shaping
- Broadcast Storm Control
- Congestion control on all ports
- IEEE 802.1p with 8 priority queues per port for different types of traffic
- WFQ (Weighted Fair Queuing)/WRR (Weighted Round Robin)/SPQ scheduling algorithm
- DSCP to 802.1p priority mapping
- DSCP
- Fast leave
- IGMP snooping v1, v2, v3
- MVR

Secured Service-Level IP TV & VoIP Operations

The ES-3124 Series comes with proven QoS and VLAN functions to ensure smooth operation. For instance, QoS features like multi-layer (L2/L3/L4) ACL, eight priority queues, DSCP and WFQ scheduling algorithm all maximize bandwidth usage and improve network service quality, while MVR and IGMP snooping enables effective IP TV multicasting. In addition, Fast Leave eradicates streaming quality disturbance when users surf between different channels, and bandwidth control with granularity of 64 kbps brings better bandwidth administration to network operators.

Cisco IP Phone 7975G

Quality of Service (QoS) Options	Supports differentiated services code point (DSCP) and 802.1Q/p standards.

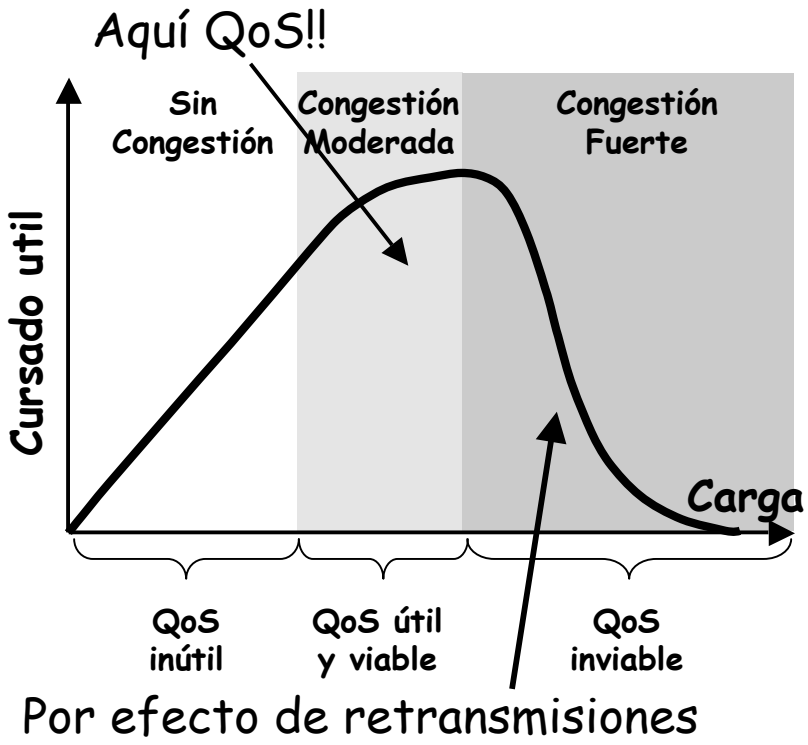
Los principios

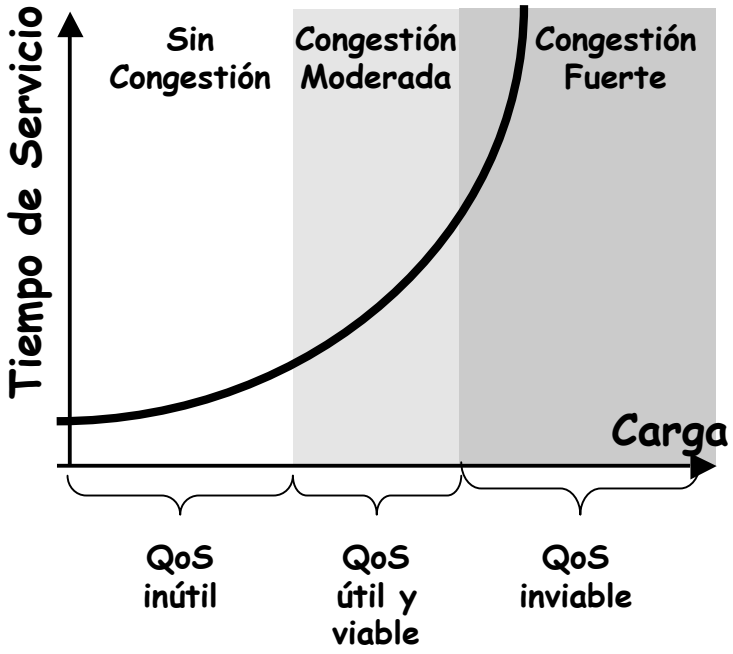
¿Qué es QoS?

- Se refiere a la habilidad de la red de *diferenciar* a unos determinados tipos de tráfico, probablemente de unos servicios concretos
- Ofrecer recursos a clases de alta prioridad *a costa de* las de baja
- Parámetros típicos:
  - Ancho de banda
  - Retardo temporal y variación en el retardo (*jitter*)
  - Probabilidad de error (o pérdida de paquetes o fiabilidad)
- Directamente relacionado con:
  - Tamaño de colas
  - Congestión de la red
  - Velocidad de conmutación
  - Ancho de banda de los enlaces
- QoS provee mejores y más predecibles servicios a la red mediante:
  - Soporte de ancho de banda dedicado
  - Mejorando las características de pérdida de paquetes
  - Evitando y manejando la congestión de la red
  - Organizando el tráfico
  - Introduciendo prioridades de tráfico a lo largo de la red

Congestión y Calidad de Servicio

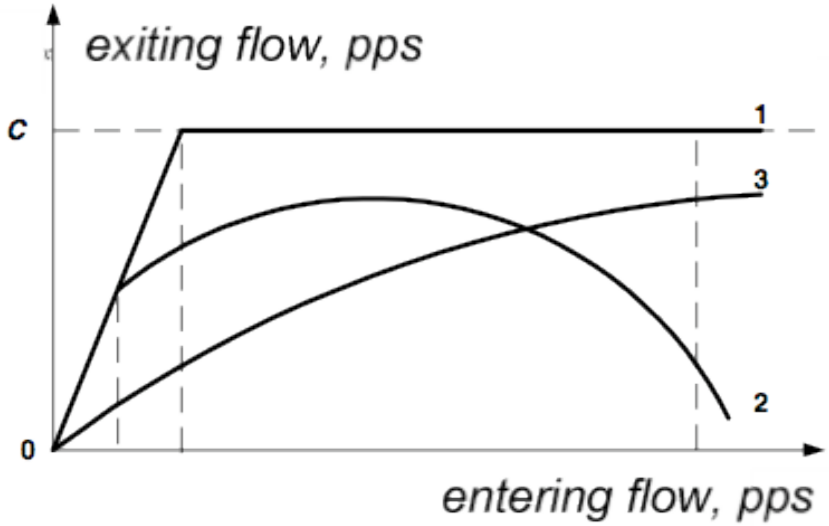
- Fácil dar QoS si nunca hay congestión
- Para ello habría que sobredimensionar todos los enlaces
- Para dar QoS con congestión es preciso tener mecanismos que permitan dar un trato distinto al tráfico preferente

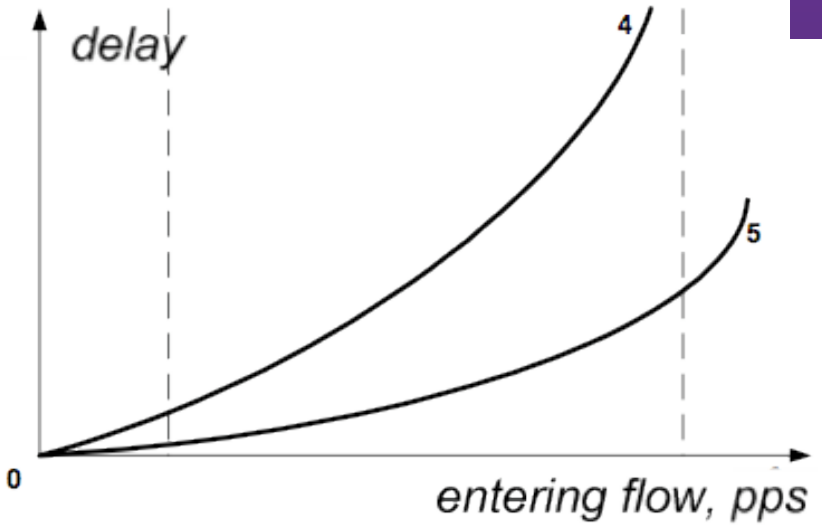




Congestión y Calidad de Servicio

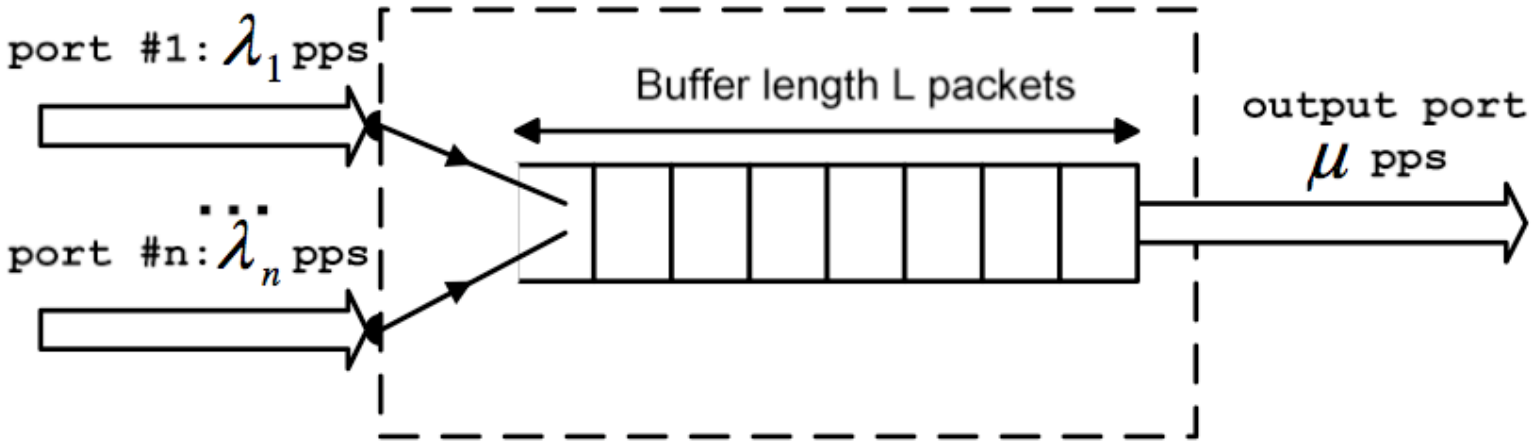
- Con y sin control de congestión
  - Flujo en el enlace y retardo en la cola
  - 1: Funcionamiento ideal
  - 2 y 4: sin control de congestión
  - 3 y 5: con control de congestión
- Control de congestión
  - Por los nodos extremo para tráfico elástico (TCP)
  - Por la red para tráfico no elástico





Servicio *Best Effort* (BE)

- Se trata igual a todo el tráfico
  - Sin separación entre flujos
  - Sin diferenciación entre paquetes
- Ante congestión
  - Crecen los retardos sin control
  - Pérdidas sin control



Ya ya, sí lo de siempre…

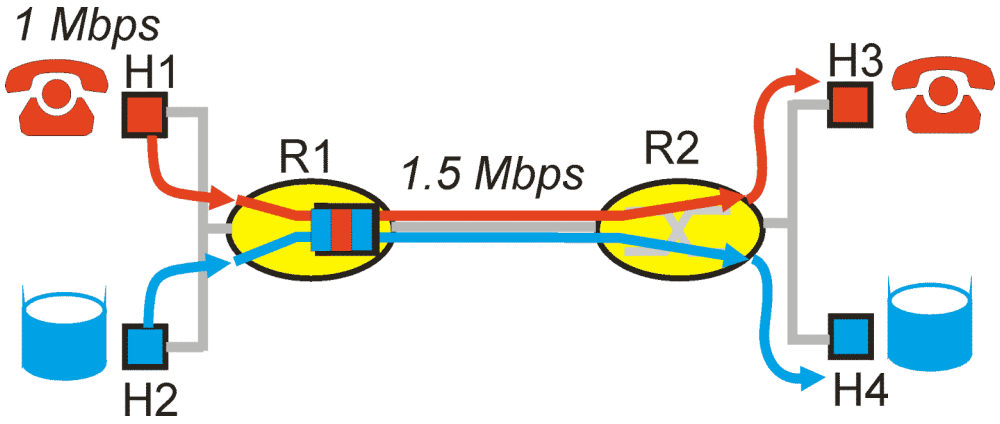
- Solución: ***Oversubscribe*** *!!*
- *Through more bandwidth at the problem !!*
- No siempre es la solución:
  - No siempre es barato aumentar el BW (ej: acceso, *peering*)
  - Si se le da más BW al usuario habrá mayor demanda
  - Un pico en la demanda degradaría la calidad de servicios sensibles
  - ¡¡ El pico puede ser de tráfico *scavenger* !! (DoS, worm… bastan 10 PCs para saturar 1GEth)
- Hoy en día ya no es suficiente para un ISP con ofrecer un servicio *Best Effort*
- La tecnología para ofrecer QoS ya es asequible y fiable
- ¡ Pero la ingeniería de estas redes no es sencilla !

Elementos

- *Connection Admission Control (CAC)*
  - *¿Puede la red cursar el nuevo flujo de tráfico manteniendo los parámetros de QoS ofrecidos a todos los usuarios?*
- *QoS routing / Traffic Engineering*
- Clasificación
  - ¿Cómo distinguir entre flujos?
- Monitorización
  - Analizar la cantidad de tráfico que entra en la red
- *Traffic shaping* y *policing*
  - Marcar, descartar o retrasar el tráfico en exceso
- 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?

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

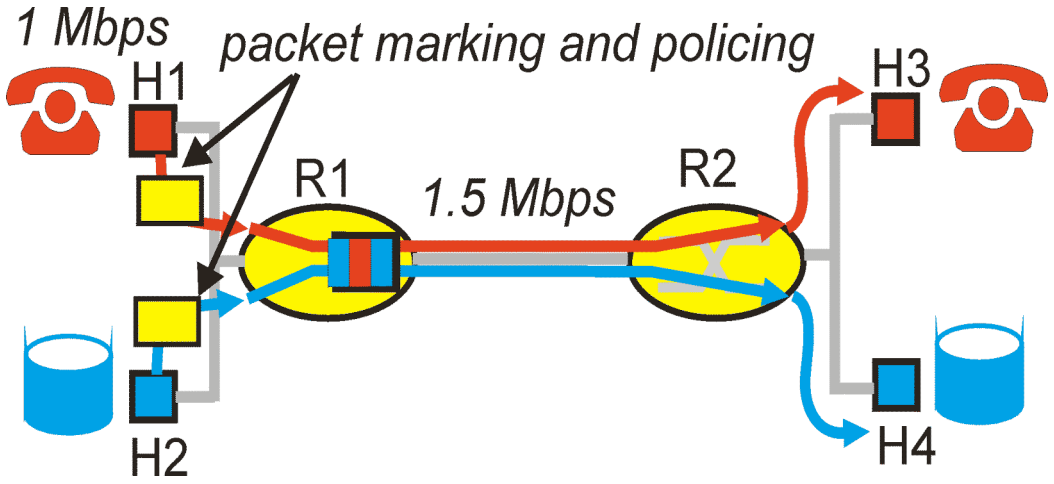


Principio 1

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

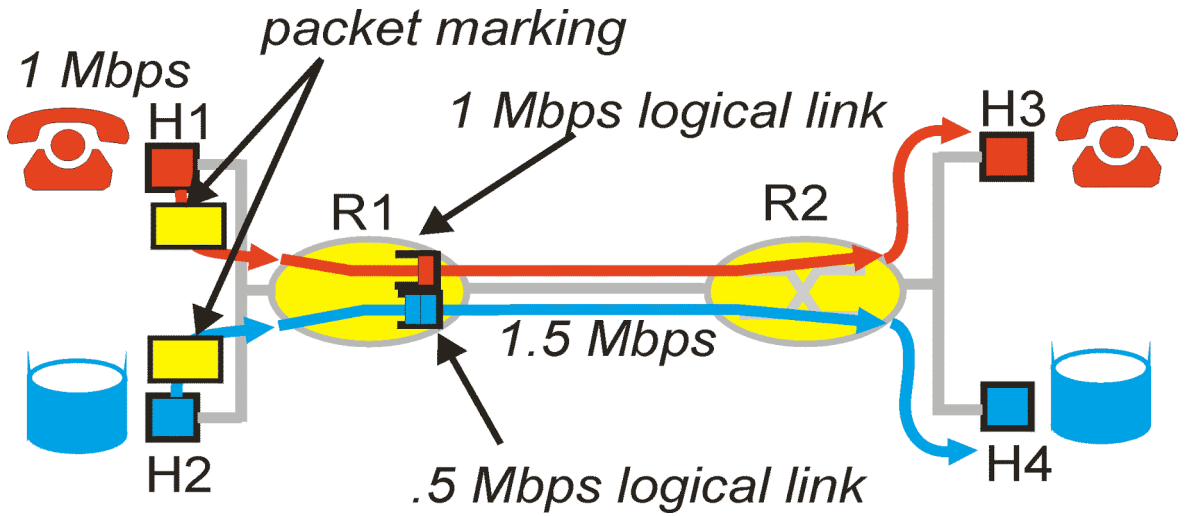


Principio 2

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

Principio 3: Eficiencia

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

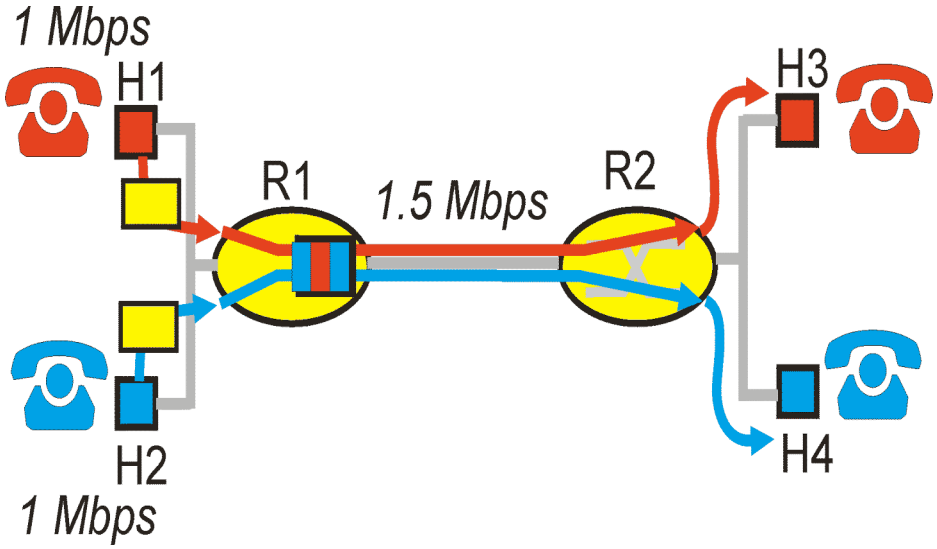


Principio 3

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

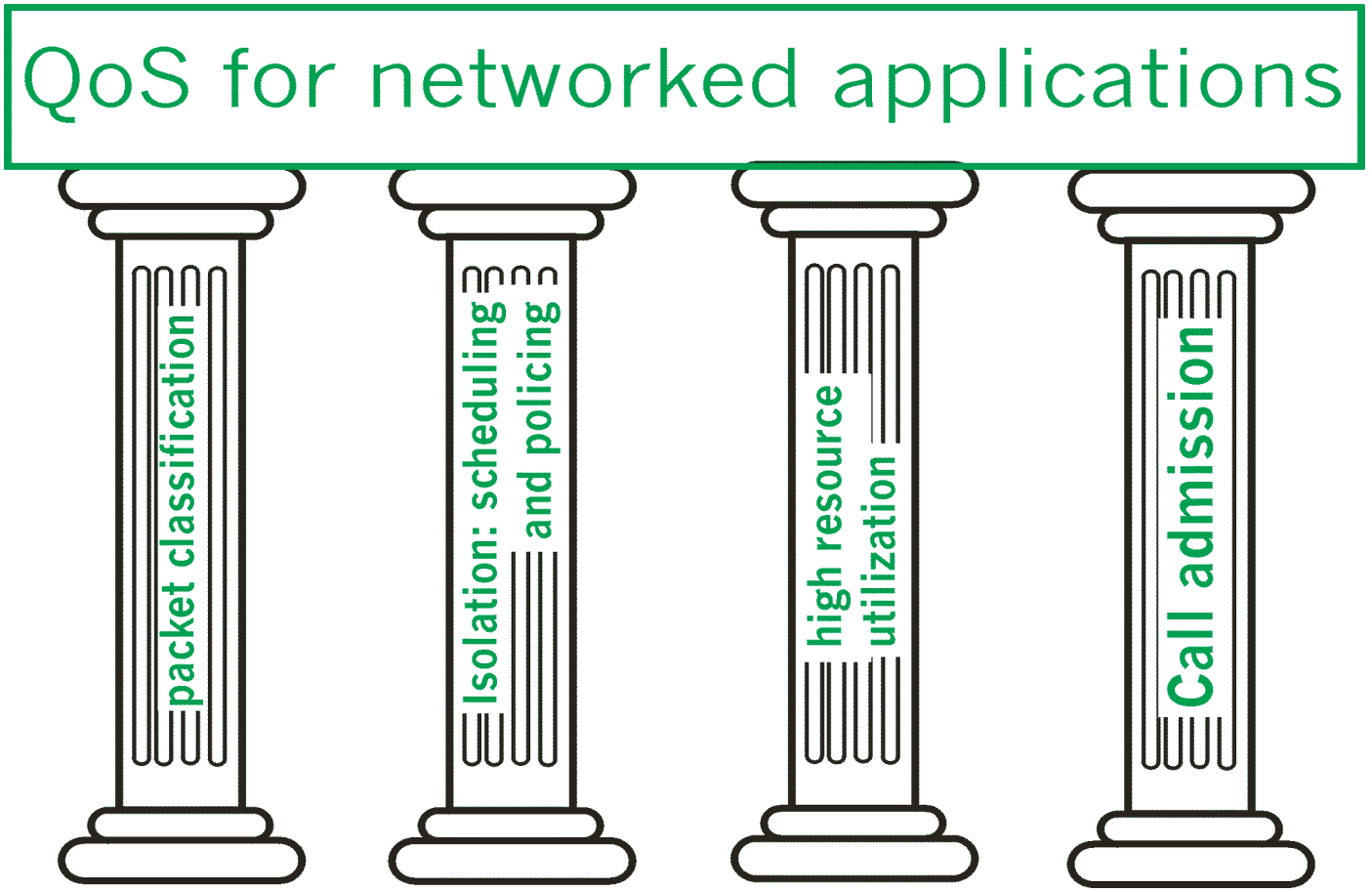
- No se pueden satisfacer las demandas más allá de la capacidad del enlace



Principio 4

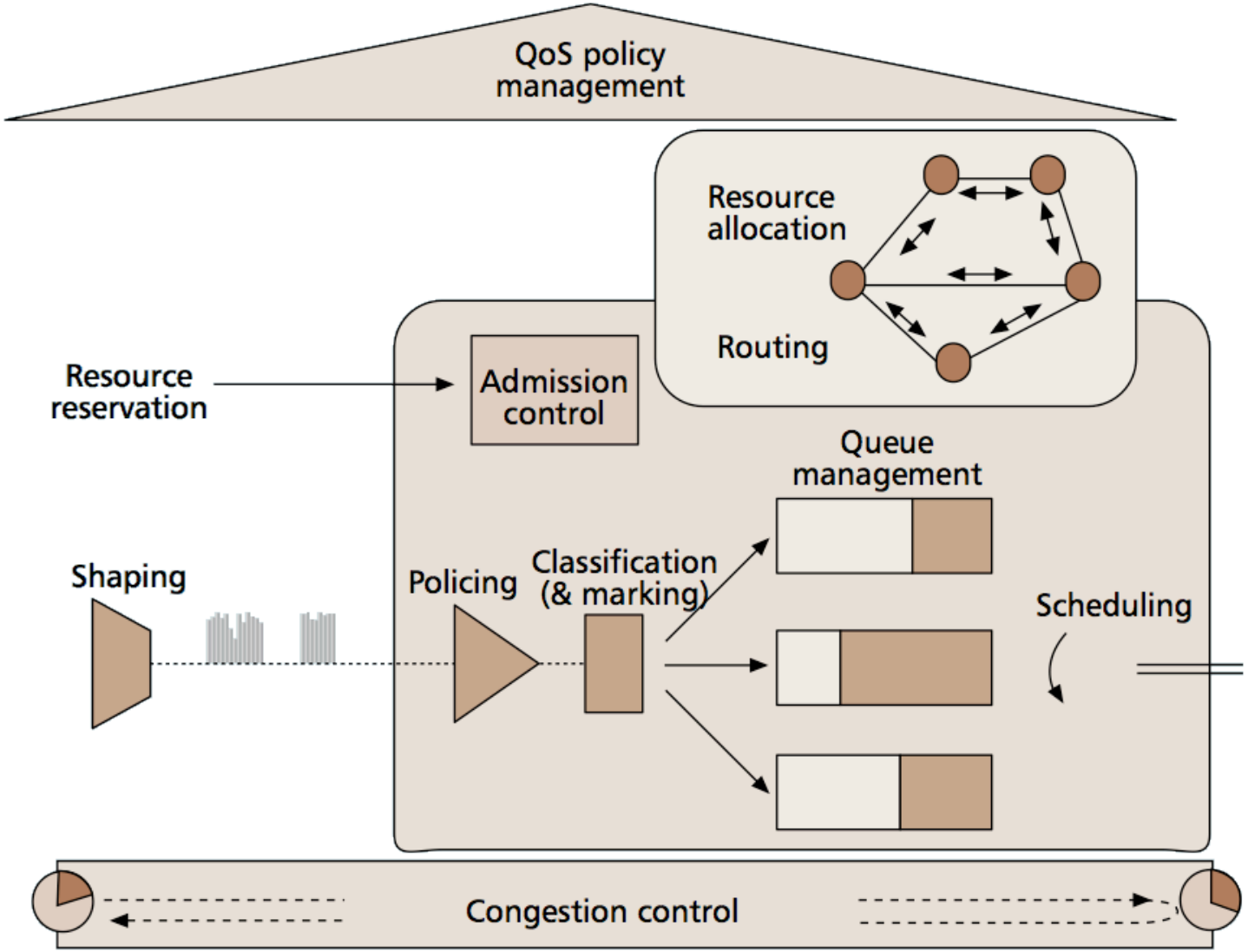
El flujo declara sus necesidades pero la red puede *bloquear* al flujo si no puede satisfacerlas: *call admission*

Summary of QoS Principles



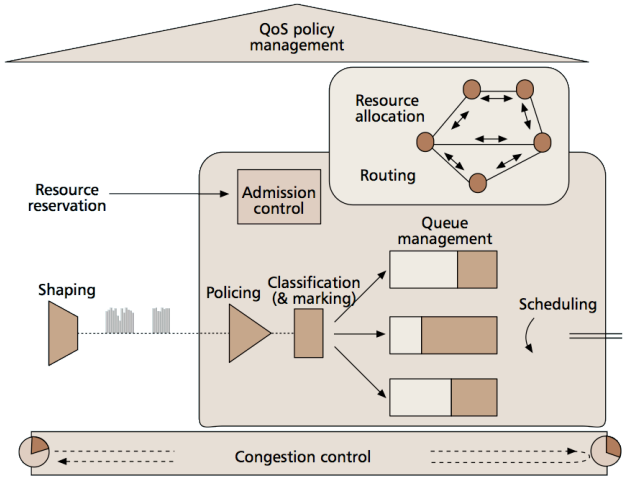
Localización de los elementos

- Router, switch o similar

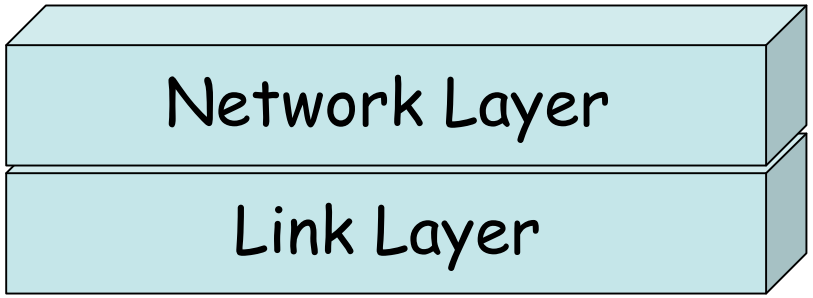


Localización de los elementos

- Router, switch o similar



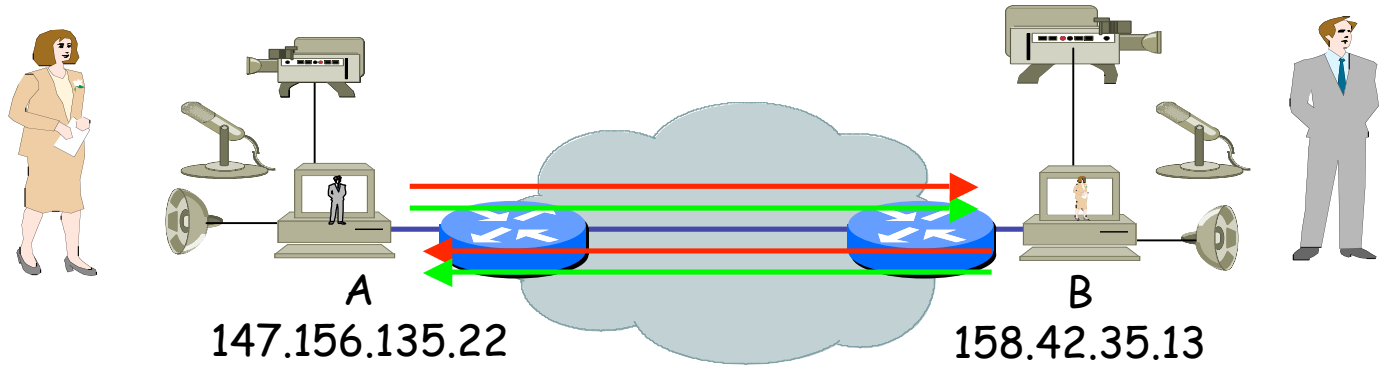
- Van a tener sentido en más de una capa



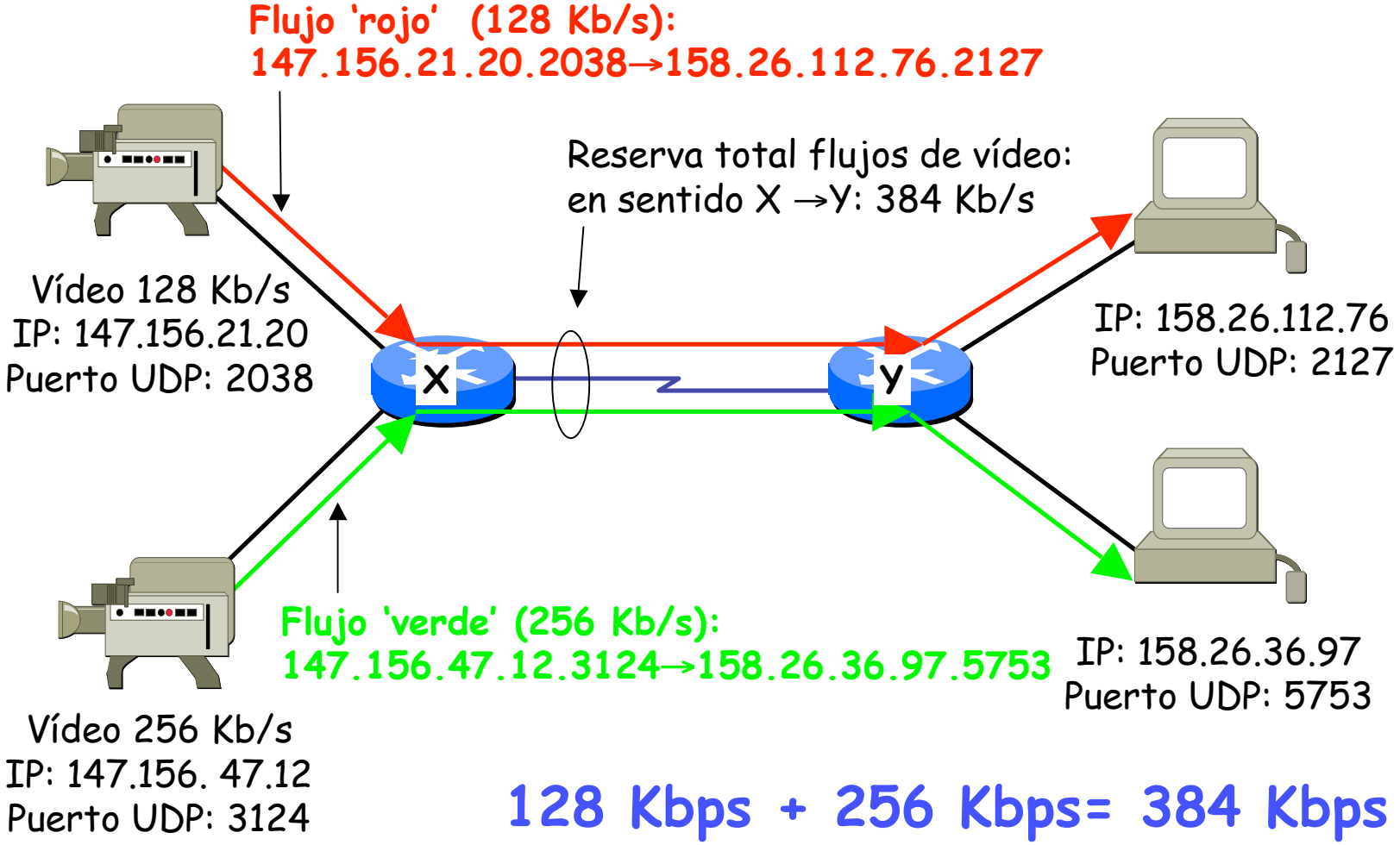
Clasificación y marcado

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.



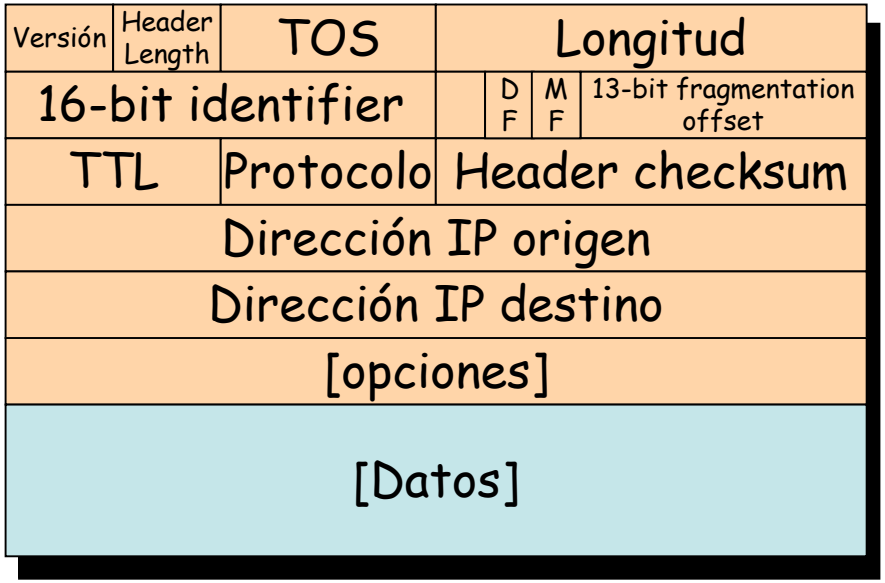
Agrupación de flujos o clases en vídeo



128 Kbps + 256 Kbps= 384 Kbps

Identificación/clasificación de flujos

- Marcar al paquete como perteneciente a un flujo
- O marcarlo como perteneciente a una clase (prioridad)
- En IPv4 (layer 3) la clasificación se hace por:
  - Dirección IP de origen
  - Dirección IP de destino
  - Protocolo de transporte utilizado (TCP o UDP)
  - Next-hop address
  - TOS

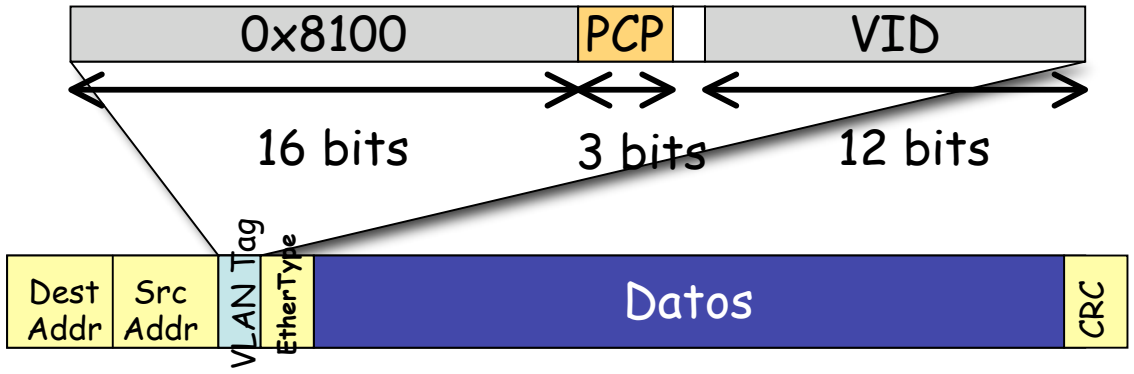


Identificación/clasificación de flujos

- En layer 4
  - Puerto origen
  - Puerto destino
- En layer 1
  - Interfaz físico por el que llega el paquete
  - PVC, subinterfaz, etc.
- En layer 7
  - Reconocimiento de la aplicación
  - URLs

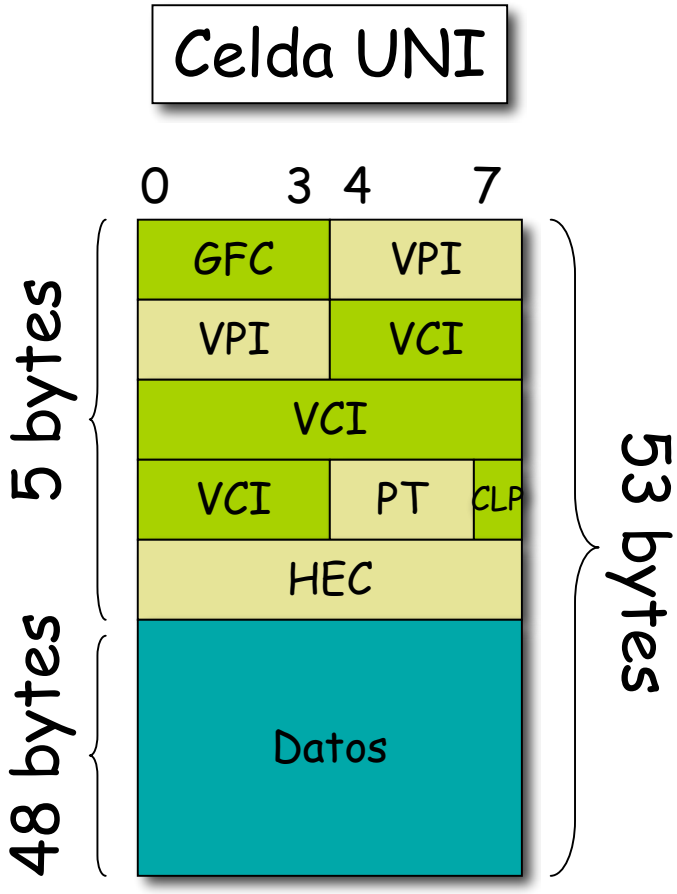
Identificación/clasificación de flujos

- En Ethernet (layer 2) se hace por:
  - Dirección MAC de origen
  - Dirección MAC de destino
  - Ethertype
  - VLAN (802.1Q)
  - Bits de prioridad (802.1p) (0=low, 7=high) (marcado)



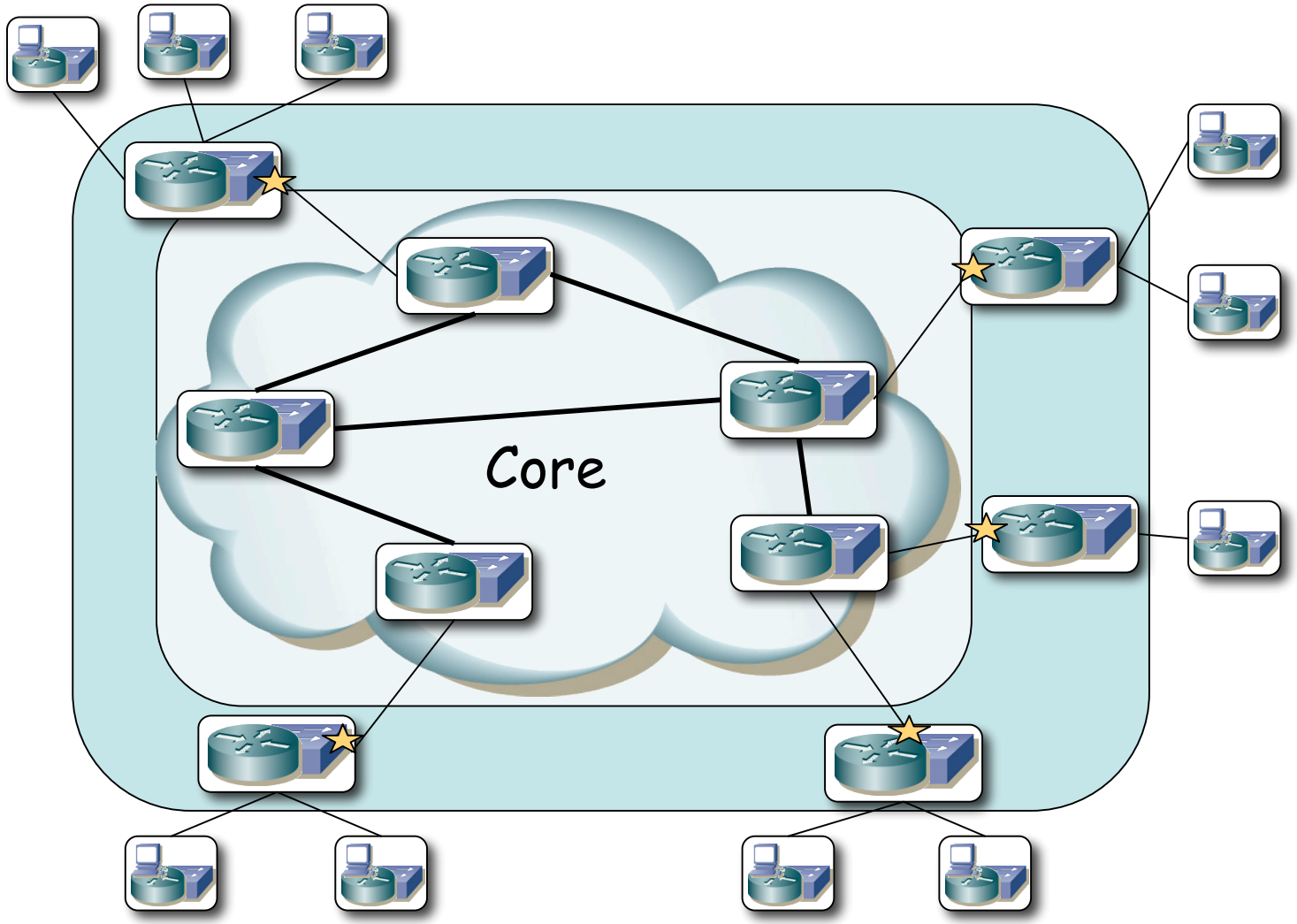
Ejemplo: Marcado en ATM

- CLP bit
  - 0 = high priority
  - 1 = low priority



¿ Dónde = Quién ?

- Preferiblemente en los extremos (edge) de la red
- O en los propios generadores de los paquetes (ej. Teléfono IP)



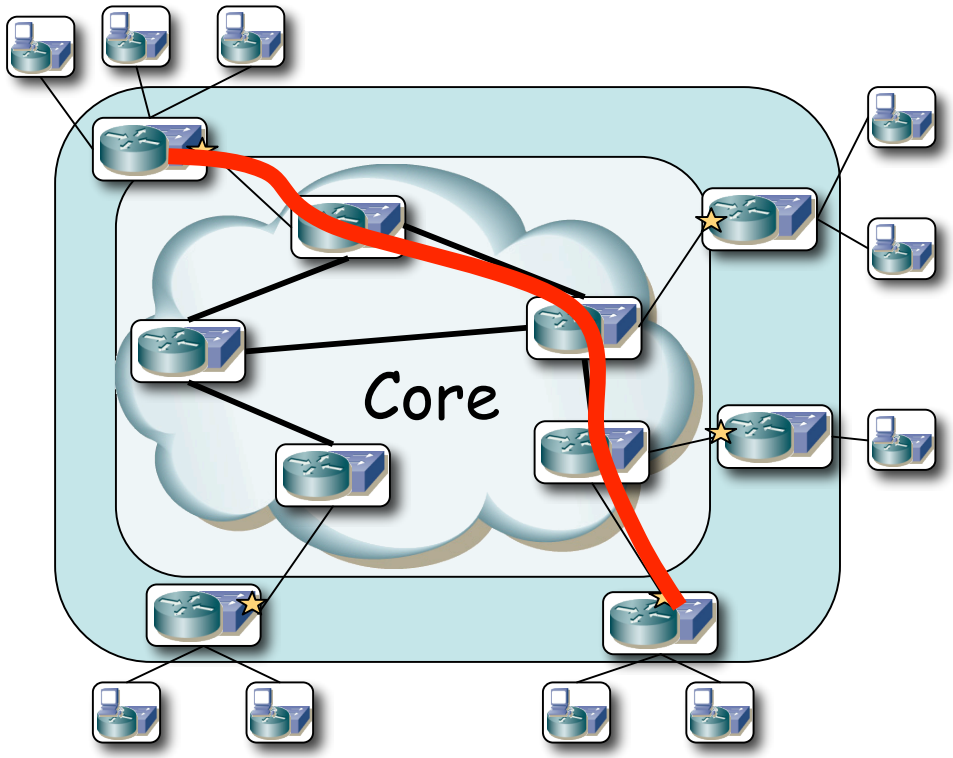
Connection Admission Control

CAC

Connection Admission Control

- Durante el establecimiento de la conexión
- Acciones para determinar si se permite o no
- Es lo correcto para flujos RT en vez de control de congestión

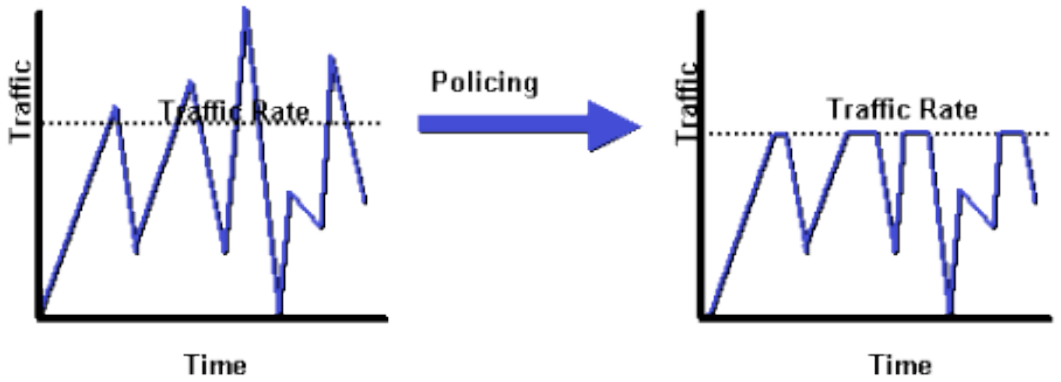
- Implica estudio de:
  - *Call routing*
  - *Preemption capabilities*
  - *Call redirection*
- ¡Proteger tráfico RT del tráfico RT!



Policing and Shaping

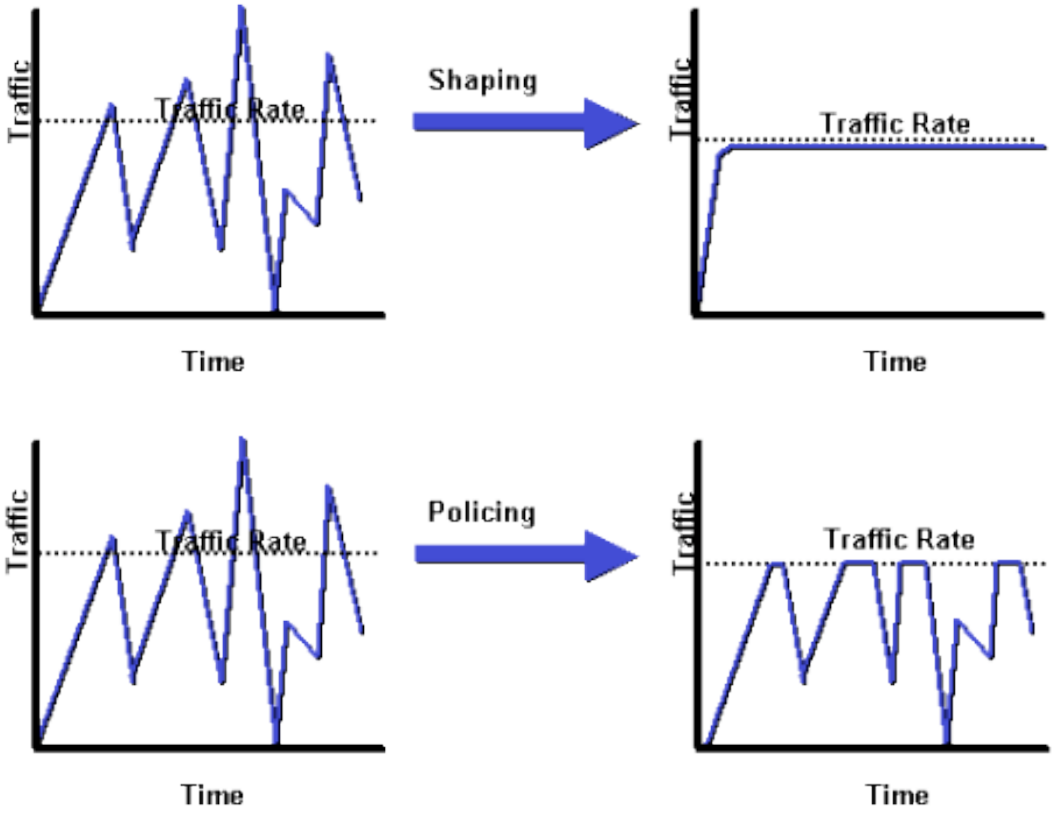
Policing

- **Objetivo**: Limitar el tráfico a la entrada a la red para que no exceda el declarado
- Características del tráfico
  - Tasa media (media a largo plazo)
  - Tasa de pico
  - Tamaño máximo de ráfaga: máx nº paquetes a tasa de pico
- Los que excedan se descartan o marcan (*conditional marker*)
- Policer as Marker:
  - Puede que el tráfico ya venga marcado según la aplicación a la que pertenezca: Separación vertical del tráfico
  - Un *policer* no distingue la aplicación, marca en función del tráfico: Separación horizontal del tráfico



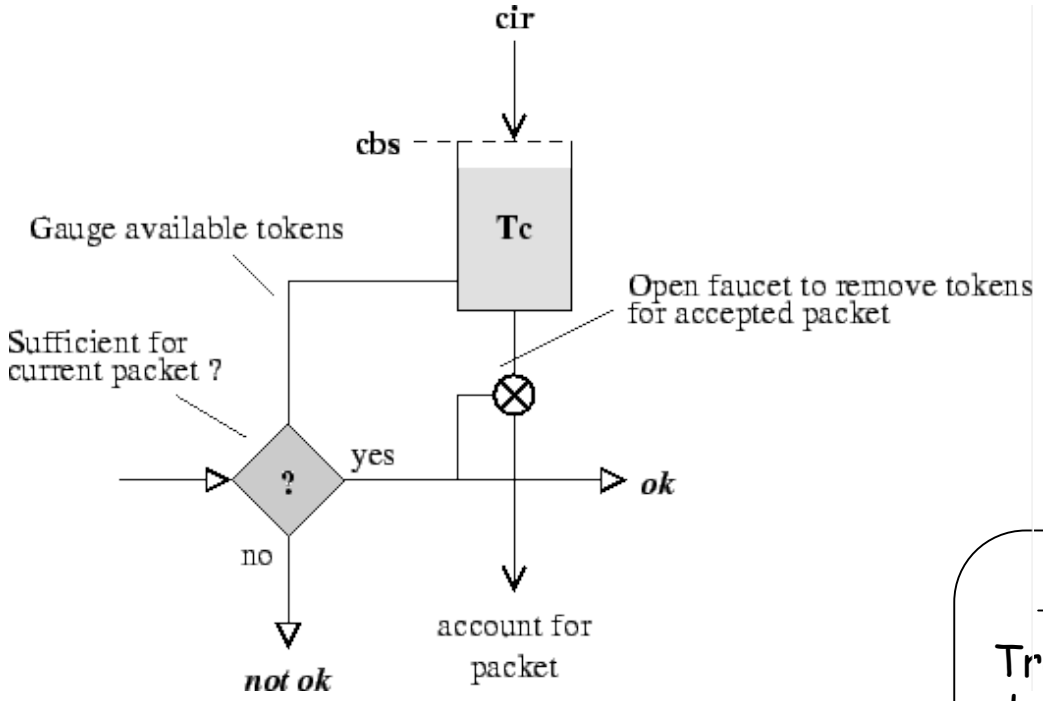
Shaping

- Los que excedan no se descartan sino que se encolan

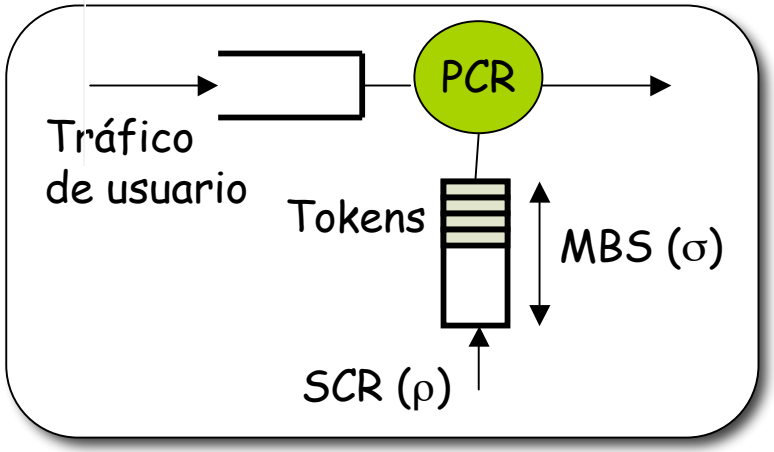


Ejemplo: *Single Leaky Bucket*

- Parámetros:
  - CIR = *Commited Information Rate* (bytes de paquetes IP por seg.)
  - CBS = *Commited Burst Size* (bytes)

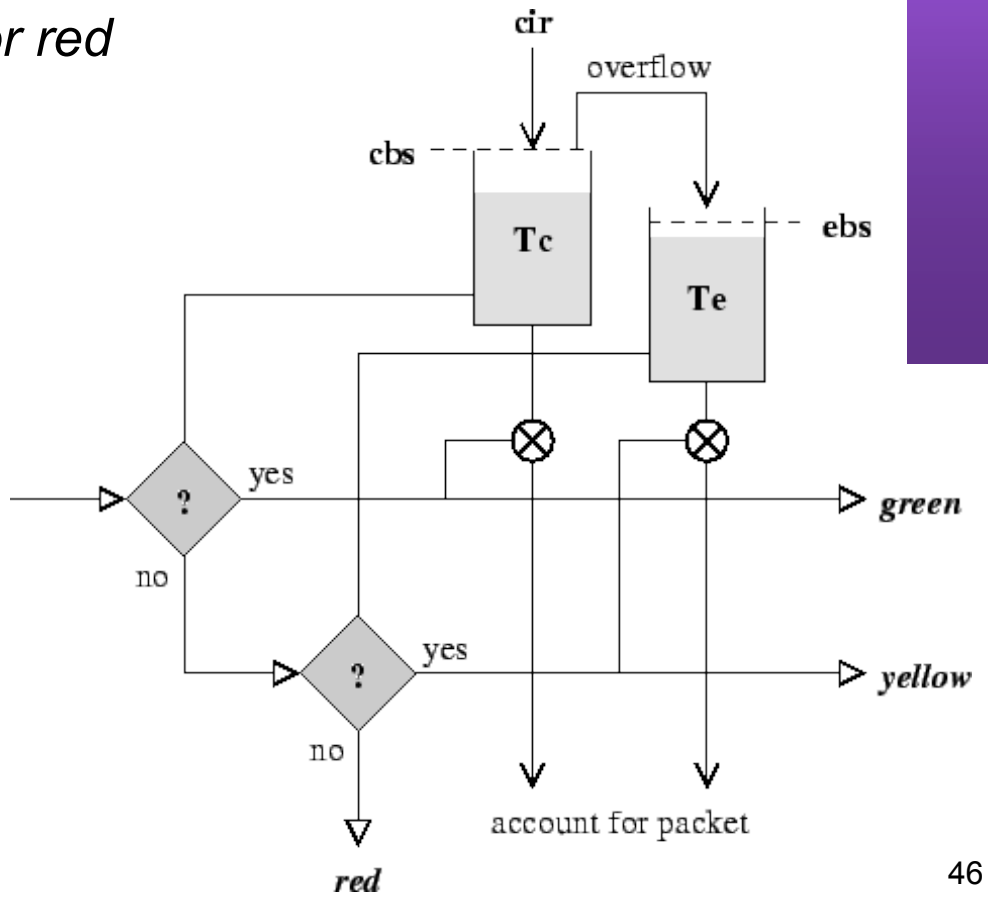


- A(0,t) = tráfico cursado en intervalo (0,t)
- $A(0,t) \leq \rho t + \sigma$
- *"Restricción (σ, ρ)"* a la salida (LBAP)



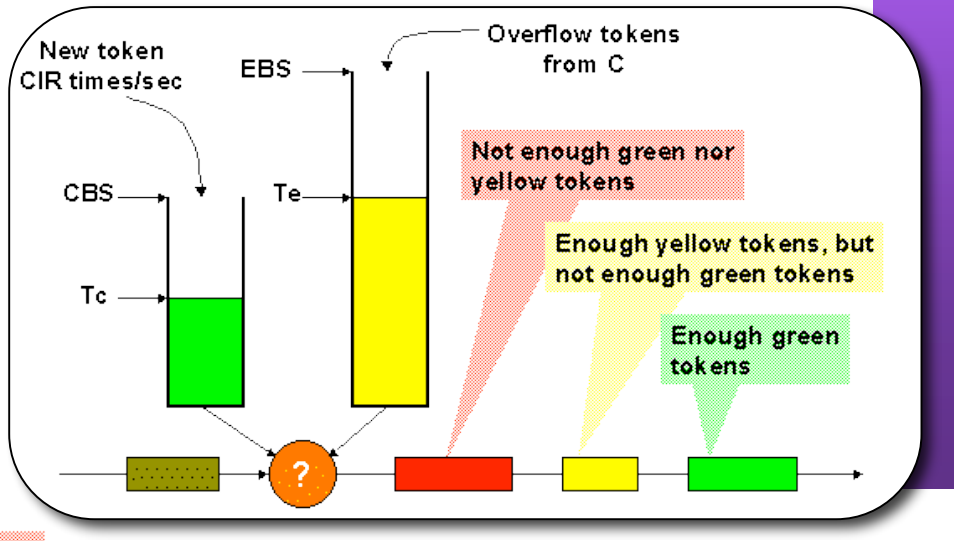
Ejemplo: srTCM

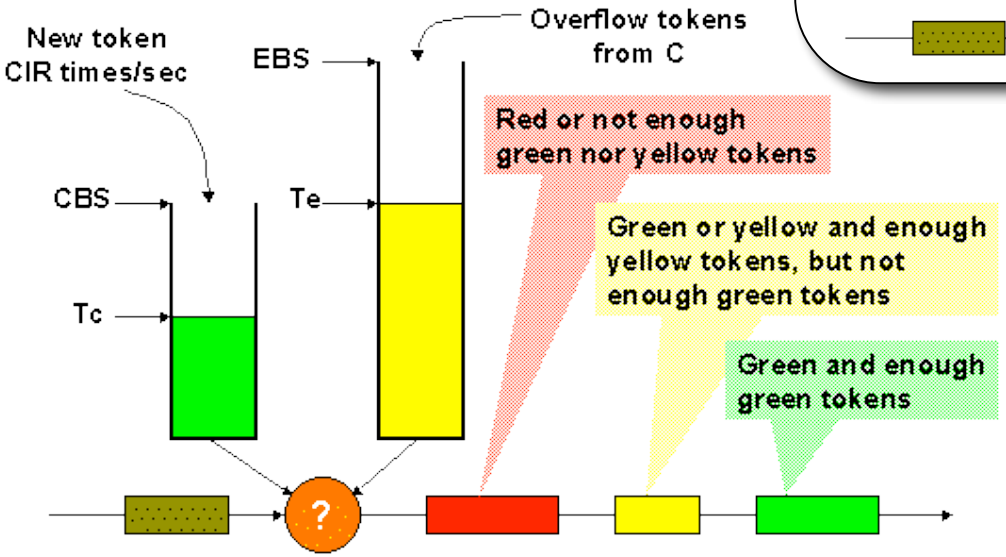
- *single rate Three Color Marker*
- RFC 2697
- Dos *Token Buckets*
- Parámetros: CIR, CBS *y EBS (Excess Burst Size*)
- Marca como *green, yellow or red*



Ejemplo: srTCM

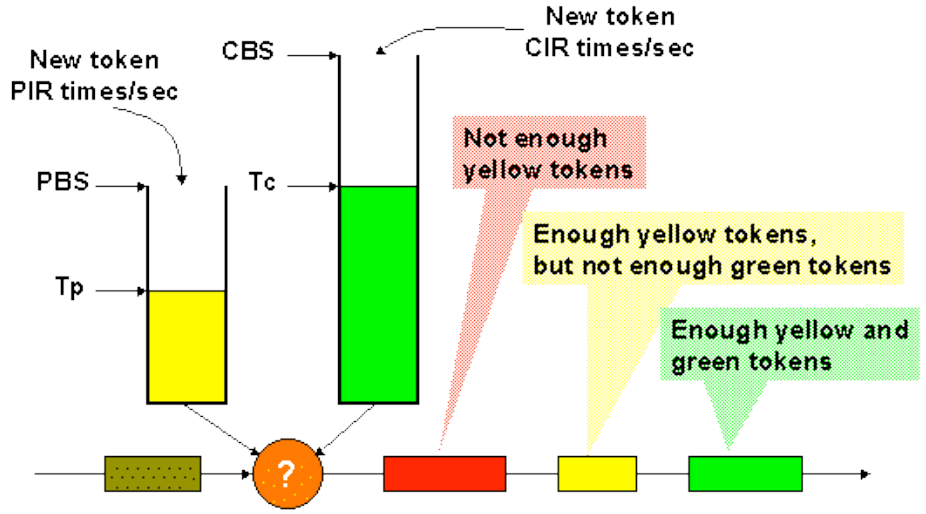
- Modo *Color Blind*:
  - If Tc(t) - B >= 0 *Green* (*Conform, DiffServ Codepoint AFx1*) else
  - If Te(t) - B >= 0 *Yellow* (*Exceed, AFx2*) else
  - *Red* (*Violate, AFx3*)
- Modo *Color Aware*:
  - If Green AND Tc(t) - B >= 0
    Green (*Conform*) else
  - If (G OR Y) AND Te(t) - B >= 0
    Yellow (*Exceed*) else
  - *Red* (*Violate*)

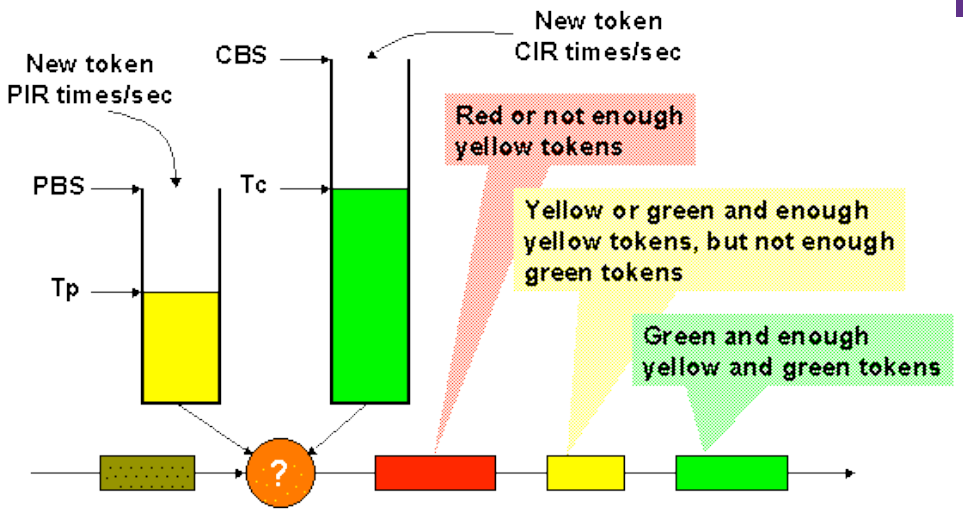




Ejemplo: trTCM

- *two rate Three Color Marker*
- RFC 2698
- PIR = *Peak Information Rate*
- PBS = *Peak Burst Size*
- *Color Blind*
- *Color Aware*
- Red : excede el PIR
- Yellow: excede el CIR





Scheduling

Scheduling

- Recursos compartidos
  - Buffer space
  - Capacidad en el enlace de salida
  - Tiempo de procesador
- Tipos de schedulers
  - Work-conserving
  - Non-work-conserving (no veremos)
- Schedulers sin prioridades
  - FCFS, RR, …
- Schedulers con prioridades
  - GPS, WFQ, SCFQ, WF2Q, …
- Características deseables
  - Sencillo de implementar
  - Reparto justo (*max-min fair share*) y protección
  - Performance bounds (deterministas o estadísticos)
  - Que permita implementar un CAC simple

The Conservation Law

- Sea un conjunto de N flujos en un planificador
- Para el flujo *i* la tasa media de llegadas es λ_i
- El tiempo medio de servicio de los paquetes del flujo *i* es x_i
- La utilización media del enlace debido al flujo *i* es $\rho_i = \lambda_i x_i$
- El tiempo medio de espera en cola de los paquetes del flujo *i* es q_i

Conservation Law

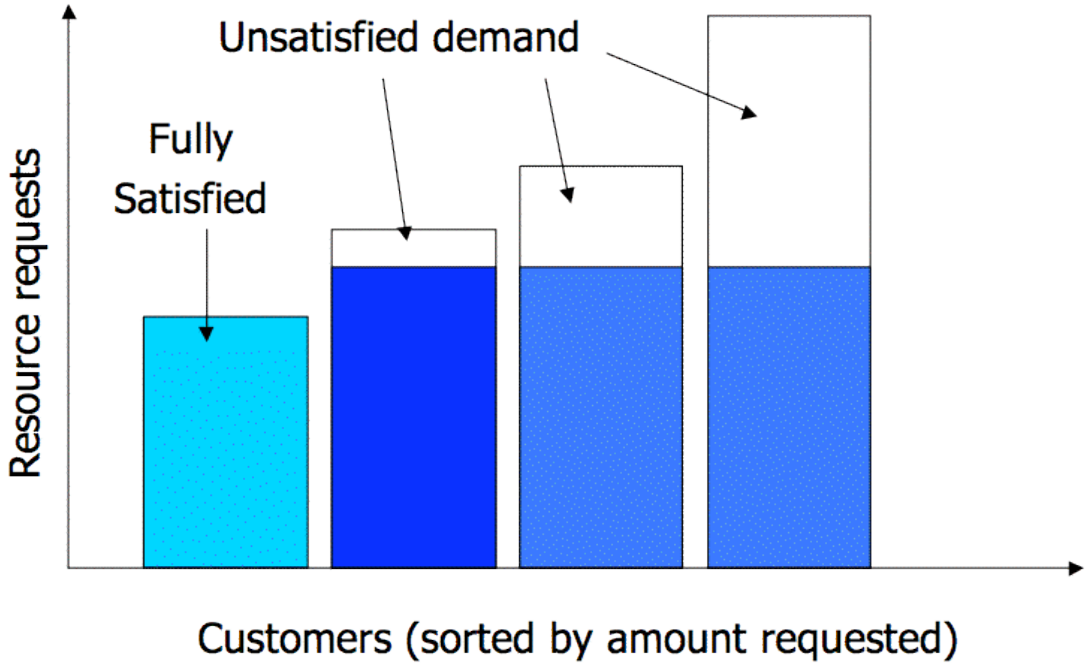
- Si el planificador es conservativo en trabajo (*work-conserving*) entonces

$$\sum_{i=1}^{N} \rho_i q_i = \text{Constante}$$

- Es independiente del planificador en concreto
- Implica que para reducir el retardo medio de una clase debemos aumentar el de otra(s)

Max-min Fair

- Asignar recursos en orden creciente de demanda
- Ningún cliente recibe más de lo que solicita
- Aquellos cuya demanda no se pueda satisfacer se reparten el remanente del recurso
- Se pueden incluir pesos

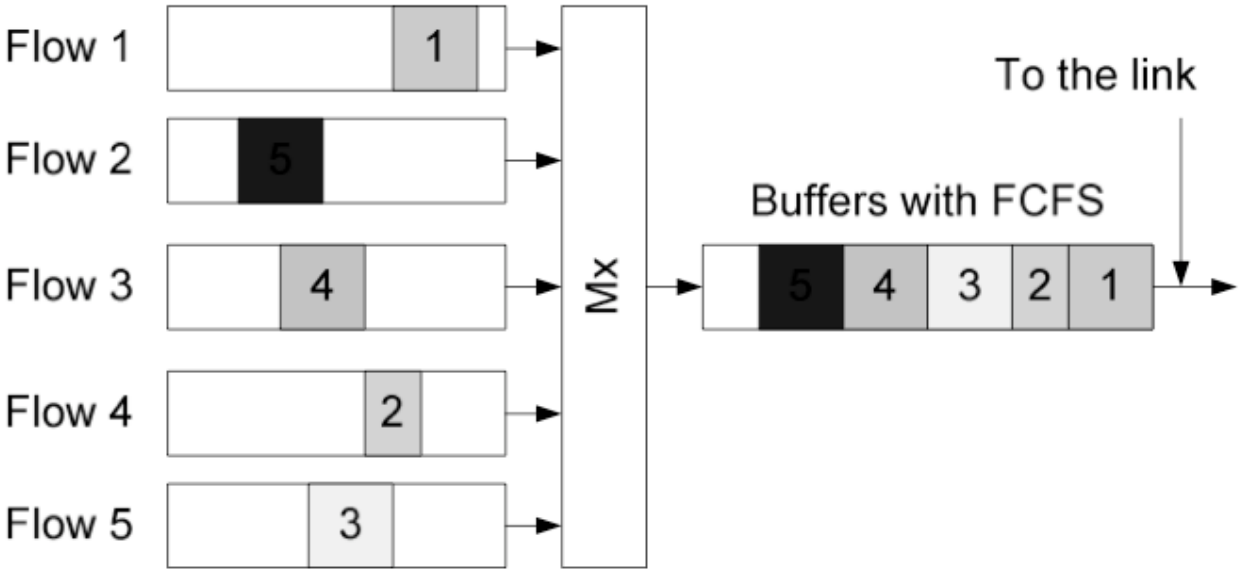


Max-min Fair (Ejemplo)

- Recurso: 10
- Demandas: 2, 2.6, 4 y 5
- 10/4 = 2.5
  - Demasiado para el primer cliente
  - Asignarle 2 y queda 0.5
- Ese 0.5 repartirlo entre los otros 3:
  - 0.5/3 = 0.167
  - Asignaciones [2, 2.67, 2.67, 2.67]
  - Demasiado para el segundo cliente
  - Asignarle 2.6 y quedan 0.07
- Repartir ese 0.07 entre los otros 2:
  - 0.07/2 = 0.035
  - Asignaciones [2, 2.6, 2.703, 2.705]

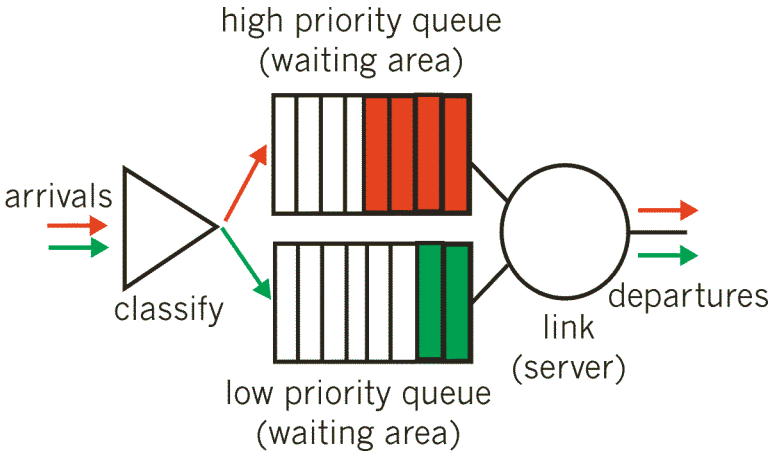
FCFS (FIFO)

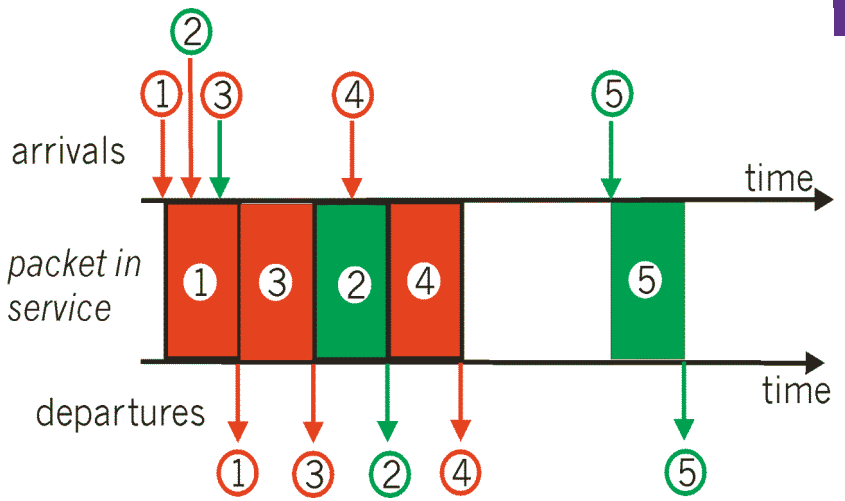
- Almacenamiento y reenvío
- Es el método más rápido y sencillo de implementar
- Se suele utilizar por defecto (B*est Effort*)
- Limitado por la capacidad del buffer ante congestión
- No permite diferenciar entre distintos tipos de paquete



Priority Queueing (PQ)

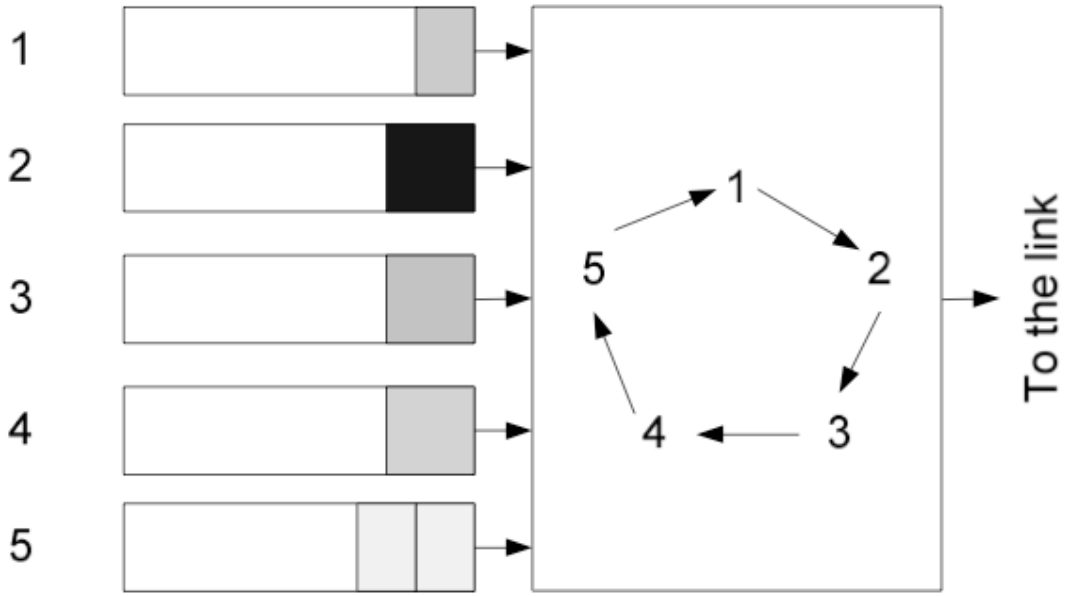
- Packets in the high-priority queue are always sent first
- Packets in the low-priority queue are not sent until all the high-priority queues become empty (*multilevel priority with exhaustive service*)
- En cada cola FCFS
- Asegura que el tráfico importante reciba un servicio rápido
- Puede crear inanición, es decir dejar fuera de servicio a tráfico menos prioritario.





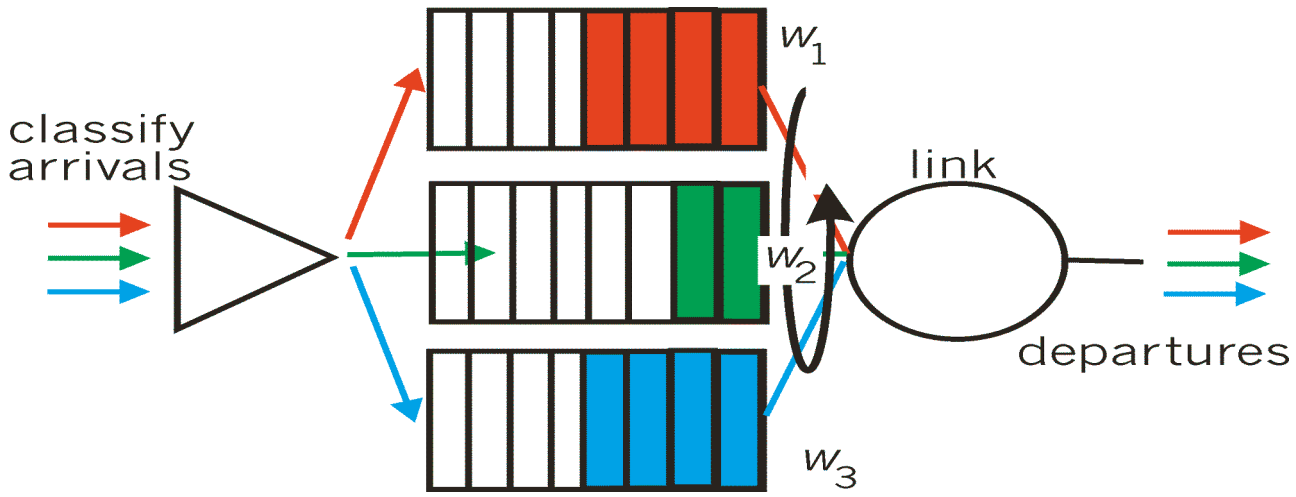
Round Robin (RR)

- Opera en “turnos” (*rounds*)
- En cada turno visita cada cola (en *round-robin*)
- En cada cola FCFS
- Se sirven un número de paquetes o paquetes durante un cierto tiempo fijo



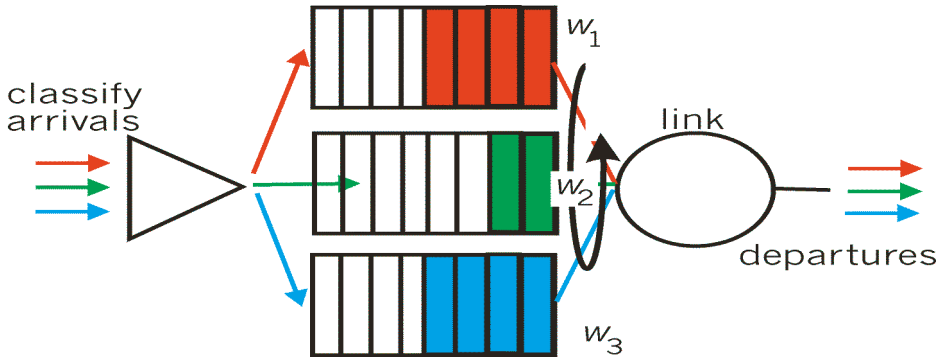
GPS

- Algoritmo basado en flujos, cada uno a una cola
- PS = *Processor Sharing* : Cada uno de los N flujos recibe 1/N
- GPS = *Generalized Processor Sharing* : pesos a cada uno
- A gran escala cada clase obtiene un servicio proporcional al peso asignado
- Asegura que las diferentes colas no se queden privadas de un mínimo ancho de banda
- No da garantías totales como PQ
- Max-min fair (y por ser *fair* ofrece protección)



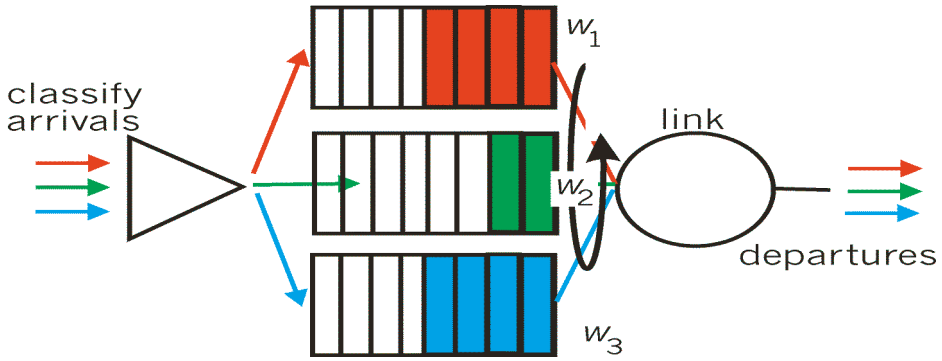
WRR

- *Weighted Round Robin*
- Aproximación de GPS (*Generalized Packet Sharing*) para el caso de paquetes
- Opera en “turnos” (*rounds*). En cada turno visita cada cola (en *round-robin*)
- Divide el peso por el tamaño medio de los paquetes del flujo
- En la visita sirve uno o más paquetes de forma que la cantidad sea proporcional al peso asignado a la cola
- *Fair* por encima de la escala del turno



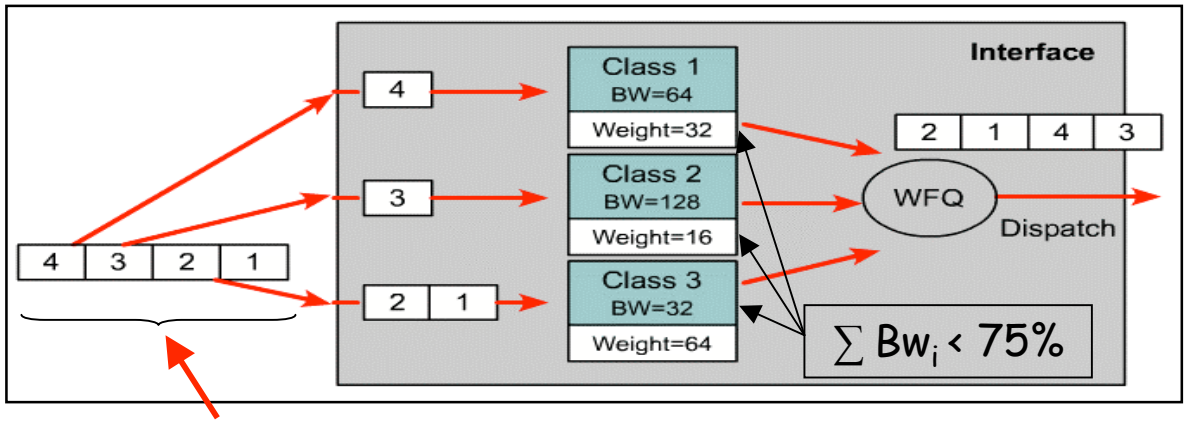
WFQ

- *Weighted Fair Queueing*
- Aproximación de GPS (*Generalized Packet Sharing*) para el caso de paquetes
- Equivalente a PGPS
- Emplea un reloj virtual
- Calcula el comienzo y final virtual en que se enviaría cada paquete en el caso ideal GPS
- Se envían en orden de tiempo final virtual
- Más complejo de implementar
- Puede ofrecer *worst-case bounds*



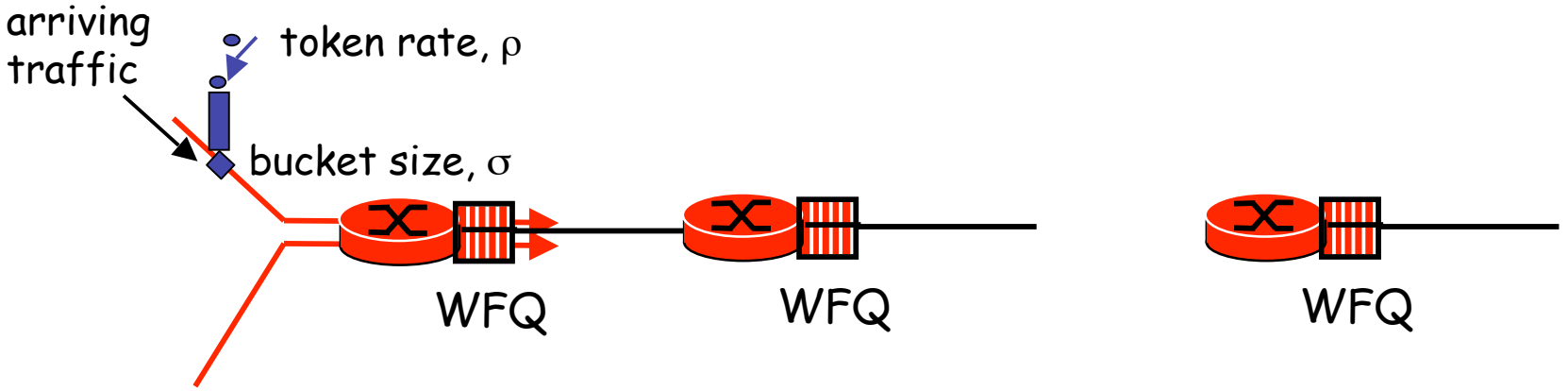
Típica implementación de WFQ

- Cada flujo es una “conversación” reconocida por info. de layer 3 (direcciones IP, precedencia) y de layer 4 (puertos)
- Pesos en función de los bits de precedencia de los paquetes
- No requiere configuración
- No escala (una cola por conversación)
- CBWFQ
  - *Class Based WFQ*
  - Especificar los filtros (clases) que determinan los paquetes que van a cada cola (una por clase, no por flujo)
  - Especificar peso para cada cola



Tocken bucket + WFQ

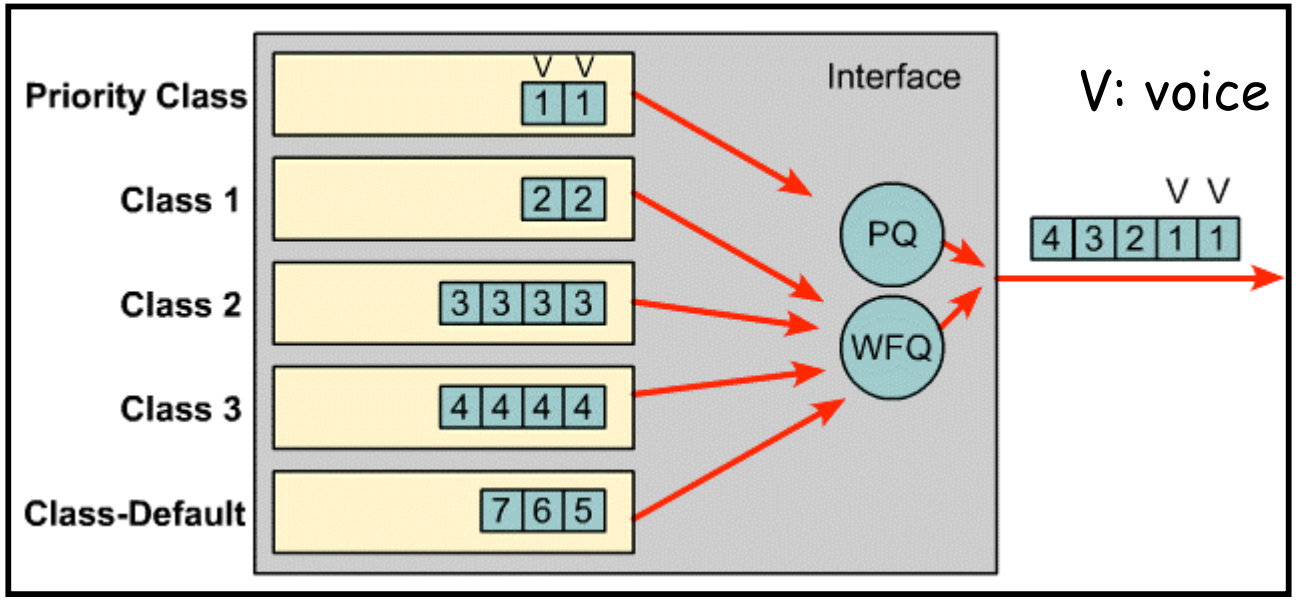
- PGPS permite ofrecer garantías de límite de retardo
- Si:
  - Flujo restricción (σ, ρ) (el resto puede no estar conformado)
  - Camino con *h* saltos (todos WFQ)
  - Se le ha asignado al menos una tasa de ρ en todos ellos
- Entonces:
  - El retardo end-to-end está acotado



Low Latency Queueing (LLQ)

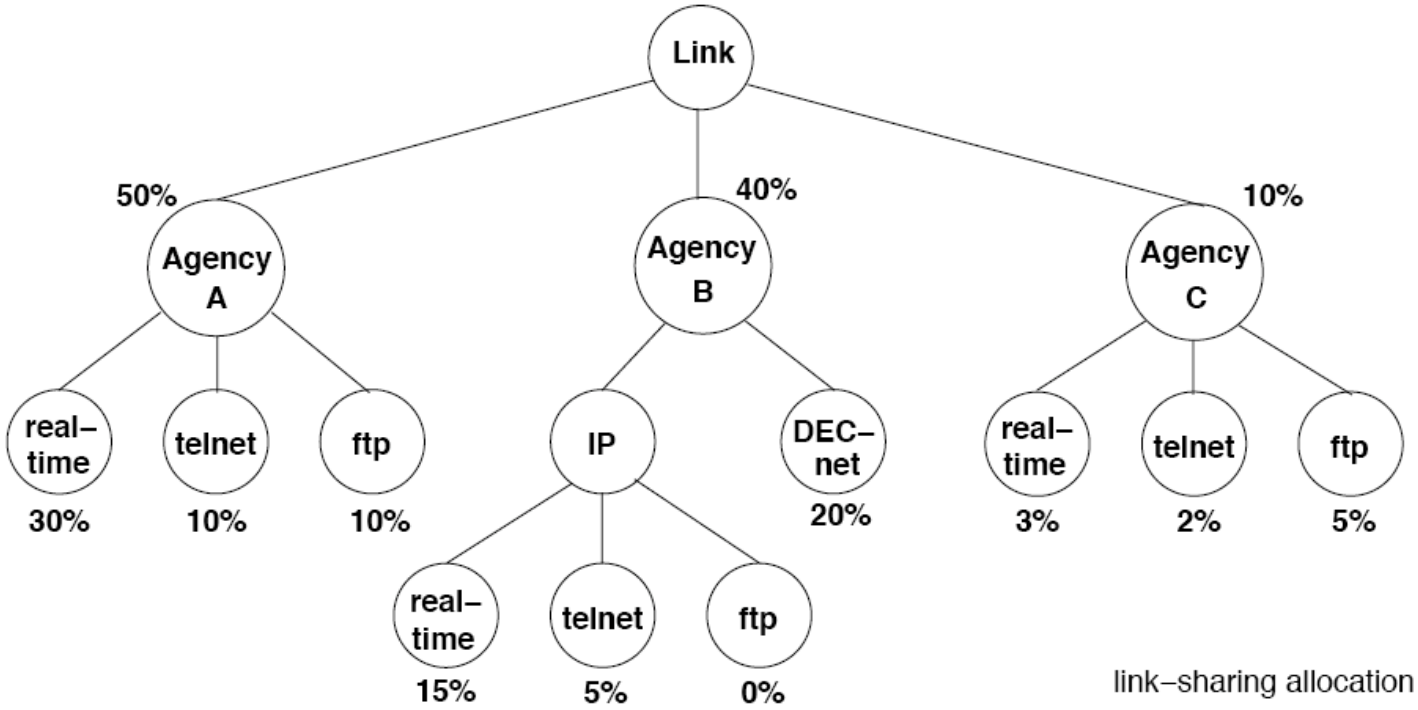
- Añade una PQ (*Priority Queue*) a CBWFQ = PQ-CBWFQ = LLQ
- Recomendable para tráfico multimedia (VoIP): bajo retardo y jitter.
- Se puede configurar junto al resto de colas CBWFQ como una cola más asociada a una clase determinada.

LLQ se comporta como una *Priority Queue*.



Class Based Queueing (CBQ)

- Puede contener diferentes planificadores

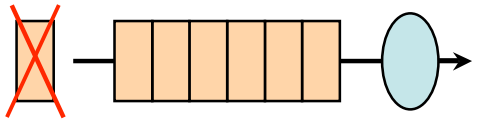


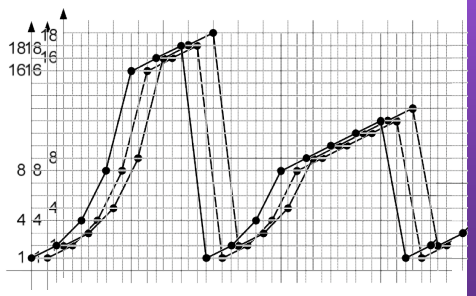
Queueing

Queue Management

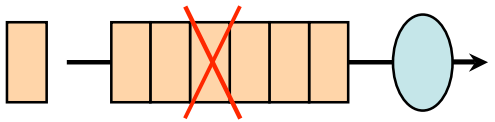
Pasivo

- *Drop-Tail* (la más habitual)
  - Simple
  - Introduce sincronización global cuando hay varias conexiones TCP atravesando ese enlace
  - Controla la congestión pero no la evita, posible synch.





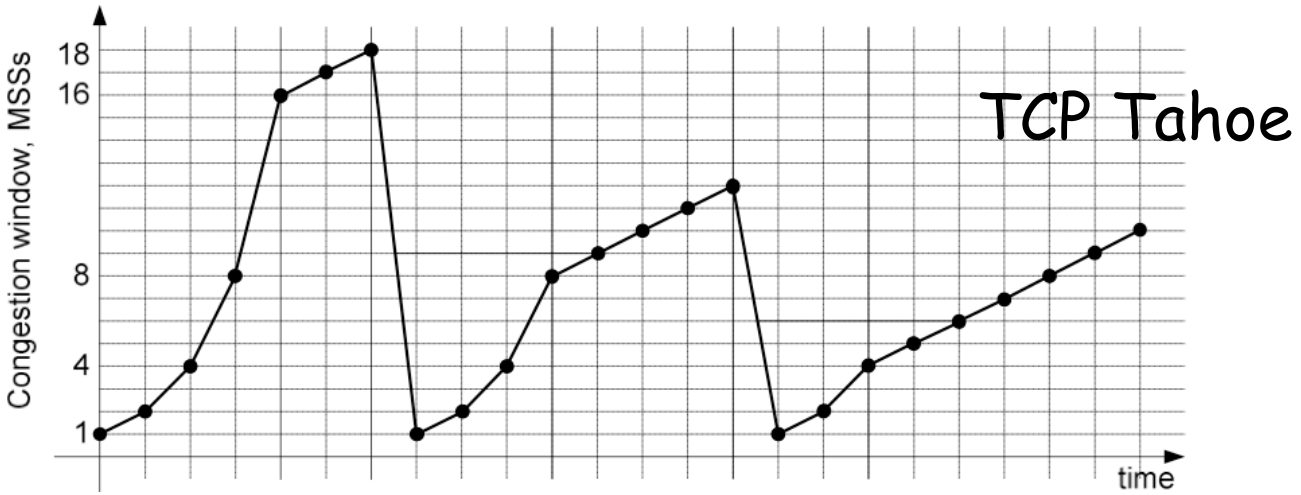
- *Head-drop*
  - Tira los paquetes que más tiempo llevan en el buffer
  - Probablemente ya han sido retransmitidos (TCP)
  - Probablemente ya llegan tarde (UDP/RTP)
  - Controla la congestión pero no la evita, posible synch
- *Random-Drop* (ante cola llena)
  - Se puede reducir la sincronización global pero no controlar UDP
  - Controla la congestión pero no la evita



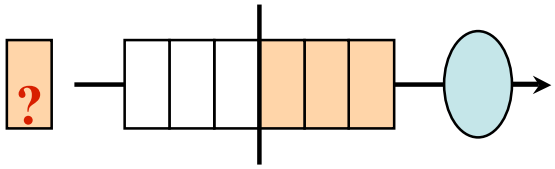
Queue Management

Activo (AQM)

- Pensando en TCP, no controla UDP igual de bien
- Evita sincronizaciones, menores retardos y fluctuaciones
- TCP regula su tasa al detectar pérdidas (*Congestion avoidance*)

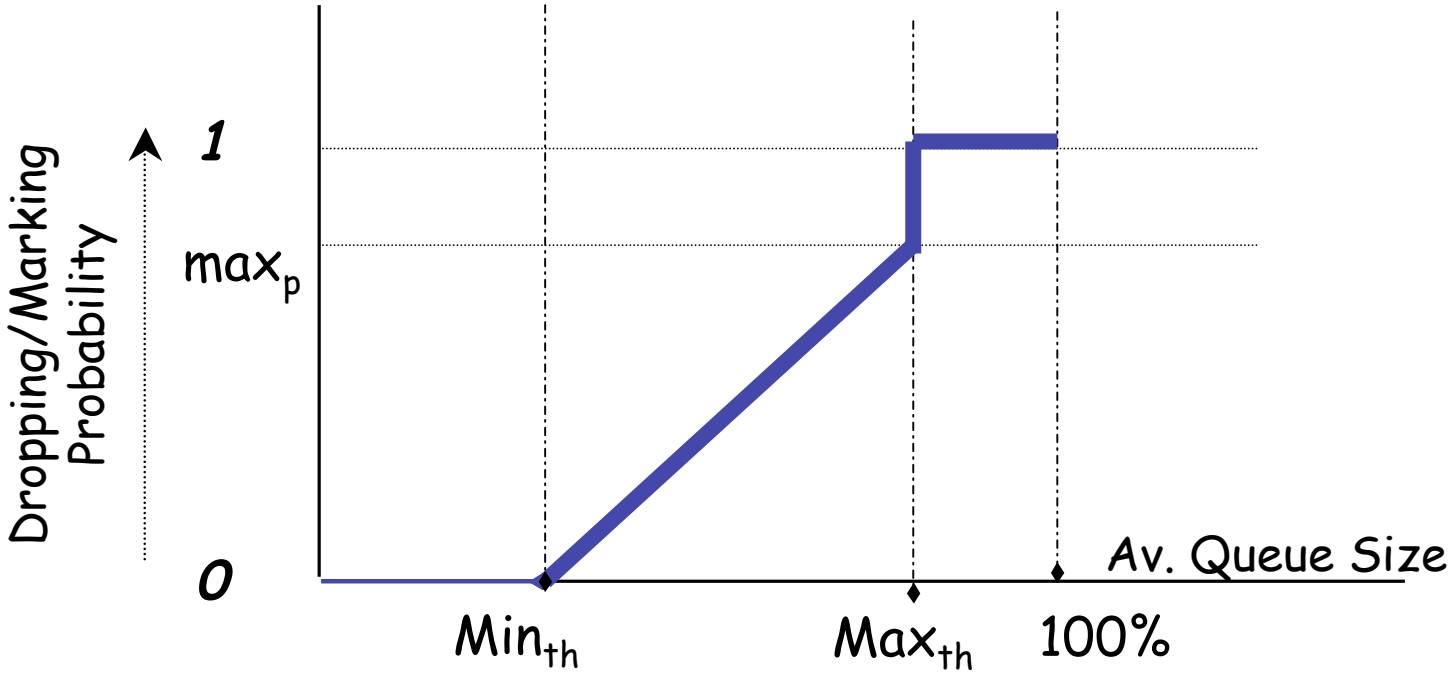


- *Early-Random-Drop* (cola no llena)
  - Si la cola excede un nivel se tira cada paquete que llega con una probabilidad fija



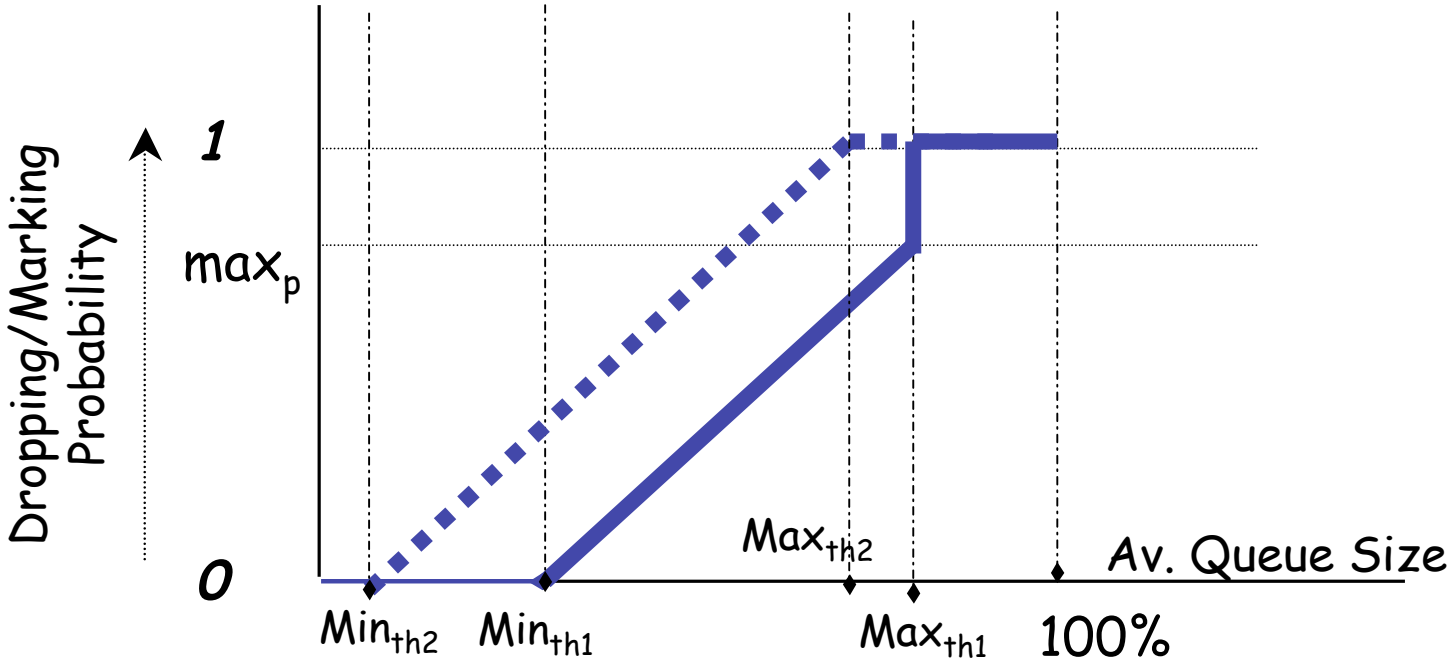
Active Queue Management

- *RED (Random Early Detection)*
  - RFC 2309
  - Evalúa la ocupación media del buffer (*exponential weighted moving average*)
  - Descartar paquetes probabilísticamente antes de la congestión
  - Ojo: Con mala configuración se comporta peor que *drop-tail*



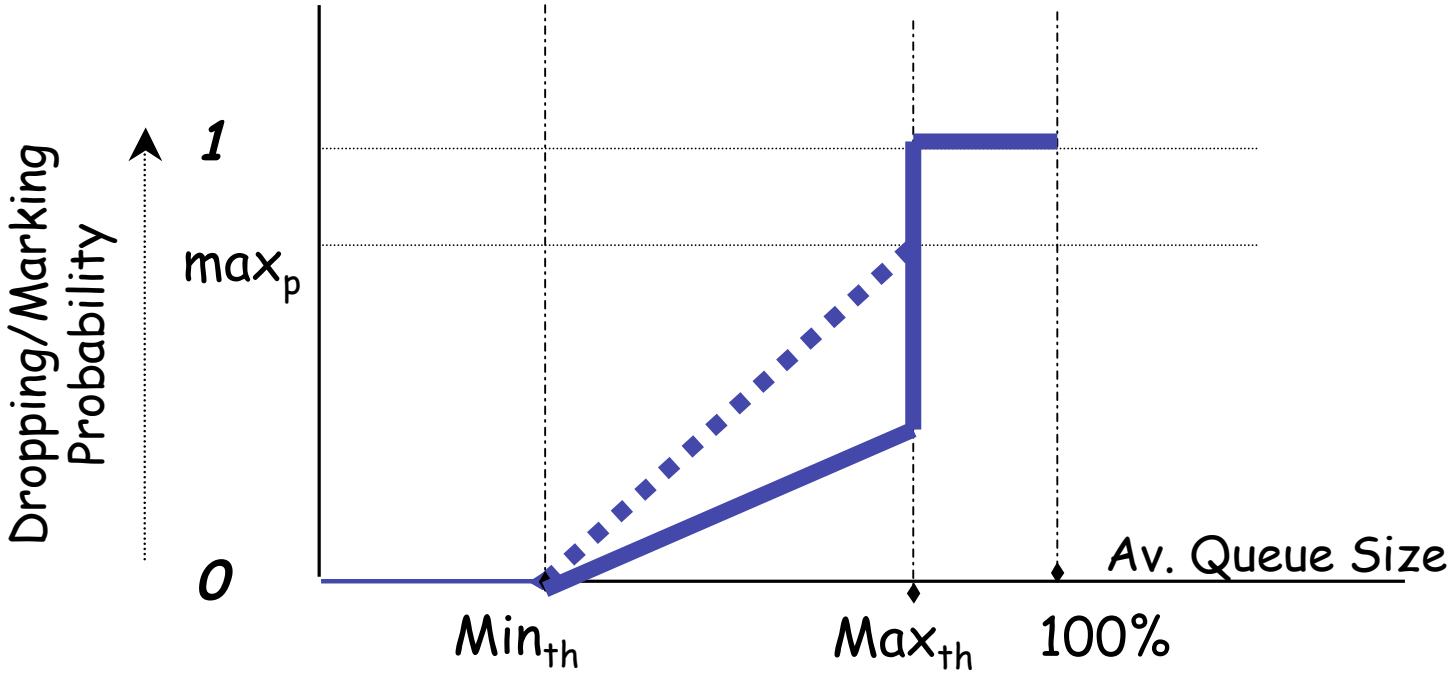
Active Queue Management

- *WRED (Weighted RED)*
  - Emplea un Min_{th} diferente para diferentes clases de tráfico
  - Mayor cuanto mayor es el valor de precedencia



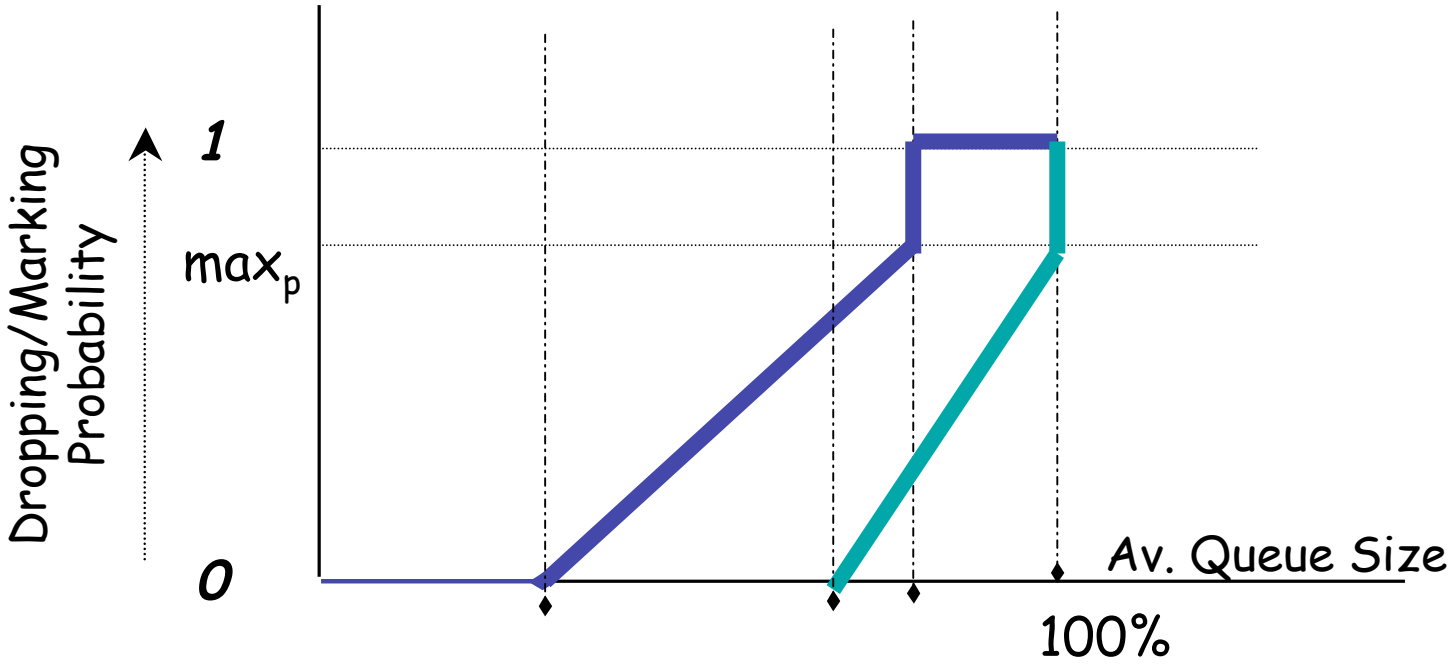
Active Queue Management

- *ARED (Adaptive RED)*
  - Que los parámetros cambien en función del tráfico
  - Difícil de implementar



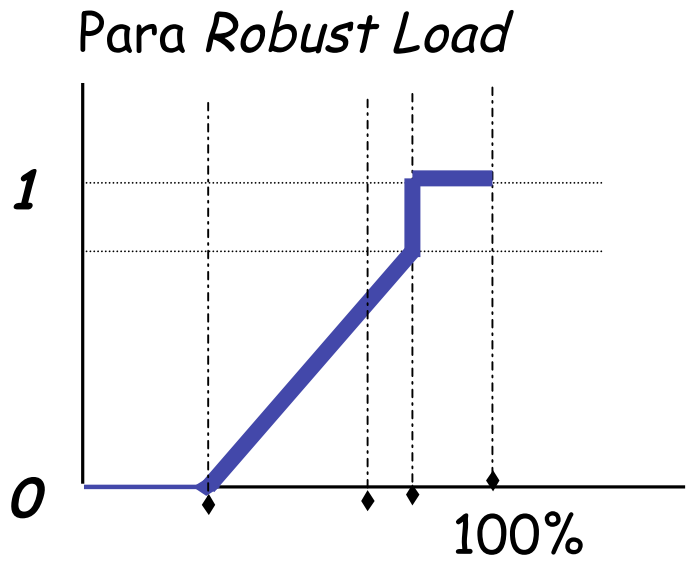
Active Queue Management

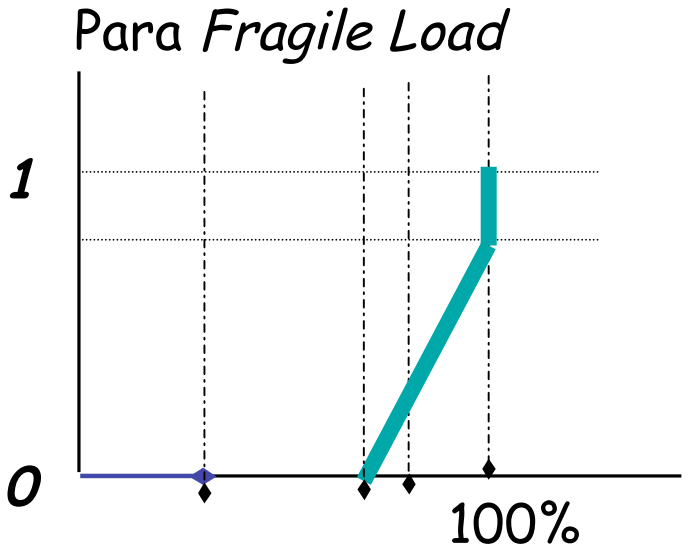
- *RIO (RED In & Out)*
  - Mantiene dos REDs simultáneos: para el tráfico *In* y el *Out*
  - Tráfico *In*: *confirming (in-profile)* (cumplen SLA)
  - Tráfico *Out*: *non-conforming (out of profile)*
  - Hay versiones en que *In* pierde solo tras entrar *Out* en p=1
  - Hay versiones en que ambos llegan a p=1 con el mismo valor de ocupación de cola



Active Queue Management

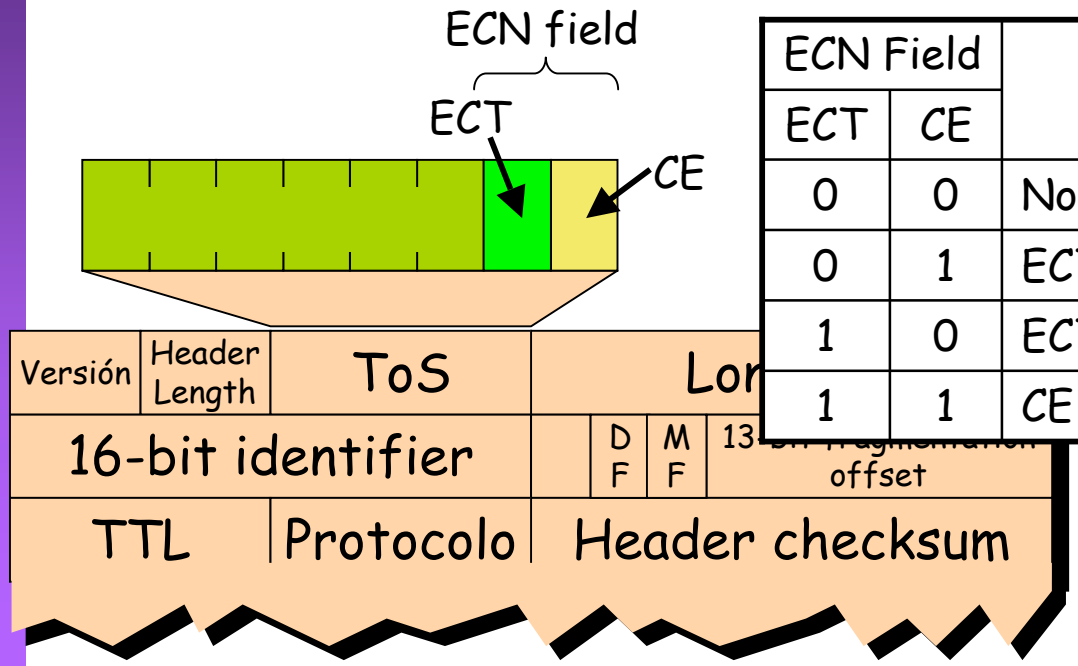
- FRED (Flow weighted RED)
  - Distinguir entre flujos
  - Carga no adaptativa (audio, video): *drop-tail*
  - *Robust Load*: TCP con pequeño RTT, reaccionan rápido
  - *Fragile Load*: TCP con gran RTT, reaccionan despacio
  - Complejo





ECN

- *Explicit Congestion Notification*
- RFC 3168
- Extensión a RED: marcar en vez de descartar
- Bit ECT = *ECN-Capable Transport*
- Bit CE = *Congestion Experienced*
- Requiere extender el control de congestión de TCP



ECN Field			
ECT	CE		
0	0	Not-ECT	Paquete no emplea ECN
0	1	ECT(1)	Extremos "ECN-capable"
1	0	ECT(0)	idem (recomendado)
1	1	CE	Congestion experienced

LFI

Link Fragment and Interleaving (LFI)

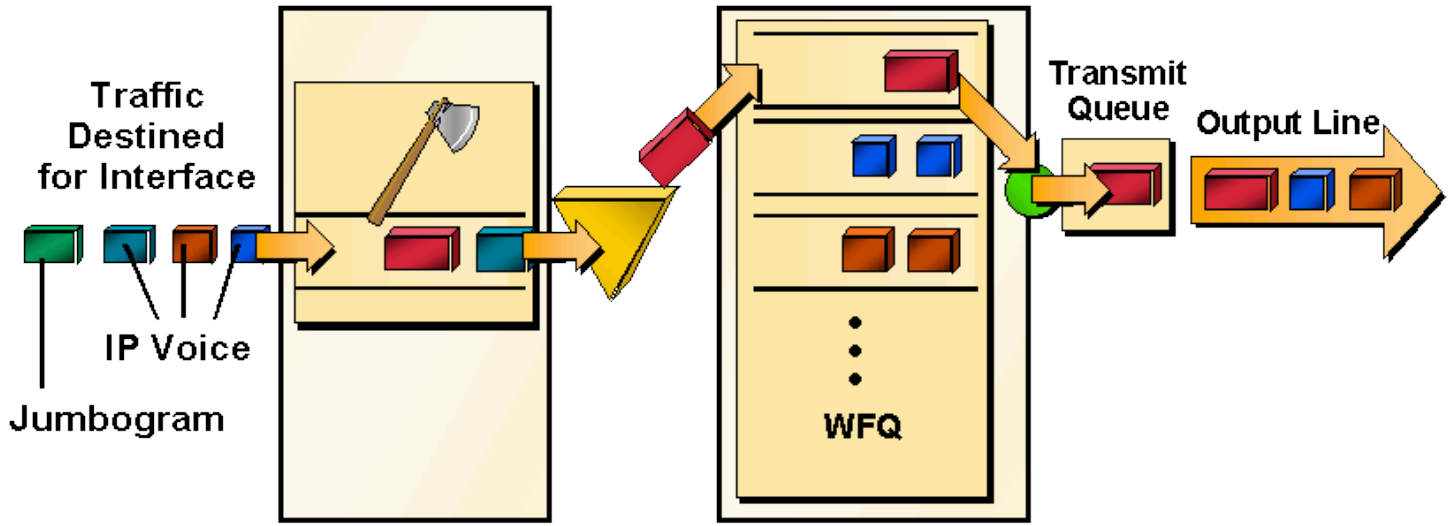
Problema:

- Llega paquete IP a su cola de alta prioridad (estando esta vacía)
- Mientras está saliendo otro paquete de clase con menor prioridad
- Retardo máximo producido si el paquete es de 1500 bytes y la línea de 256Kbps:

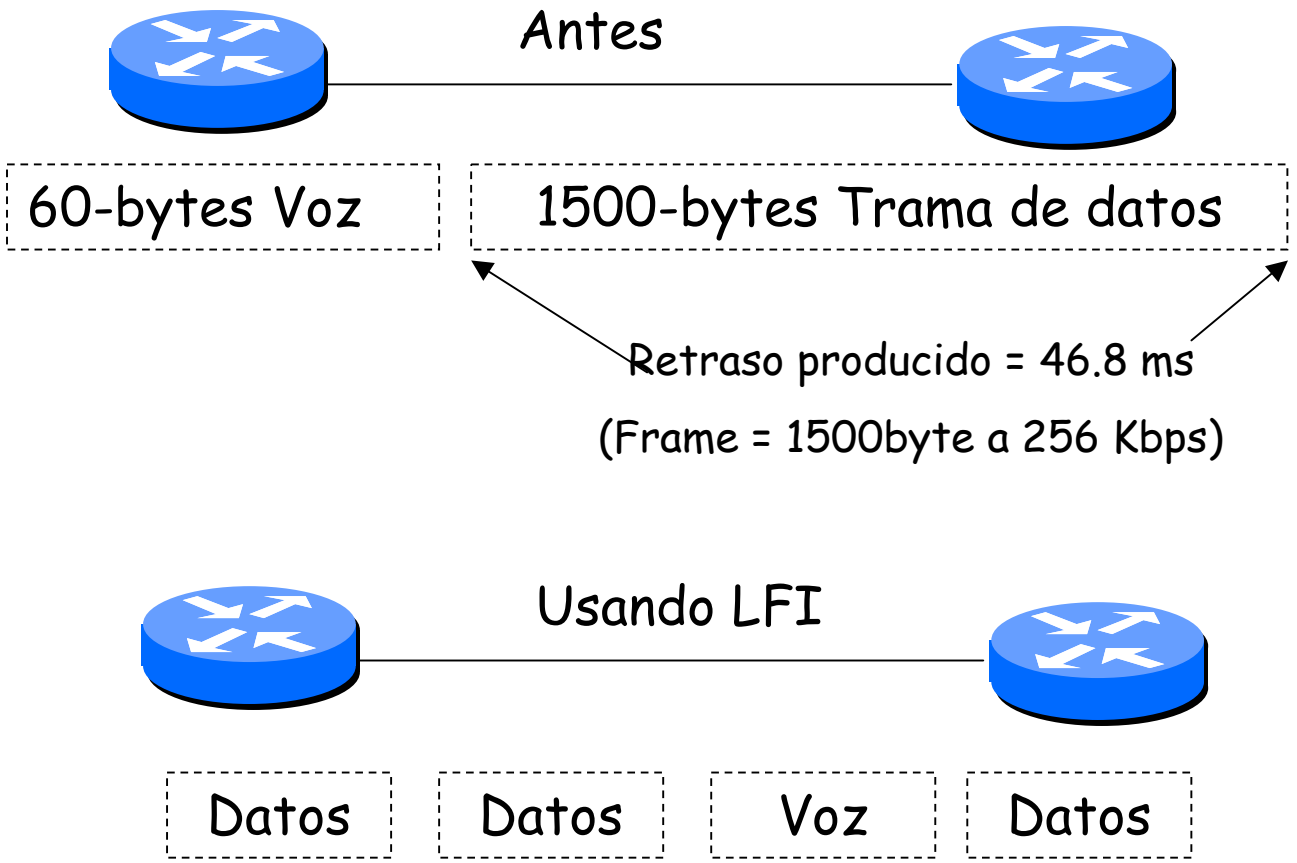
  (1500*8) bits / 256 Kbps = 46.8 ms!

Solución:

- Fragmentar los paquetes de datos
- Ej.: límite fragmentos “de 10ms”
- Es decir, tamaño de un paquete igual a máximo que se pueda enviar en 10 ms
- Insertar paquete de VoIP entre estos paquetes
- Asegura un retraso mucho menor
- ¡¡ Los paquetes VoIP no deben fragmentarse !!



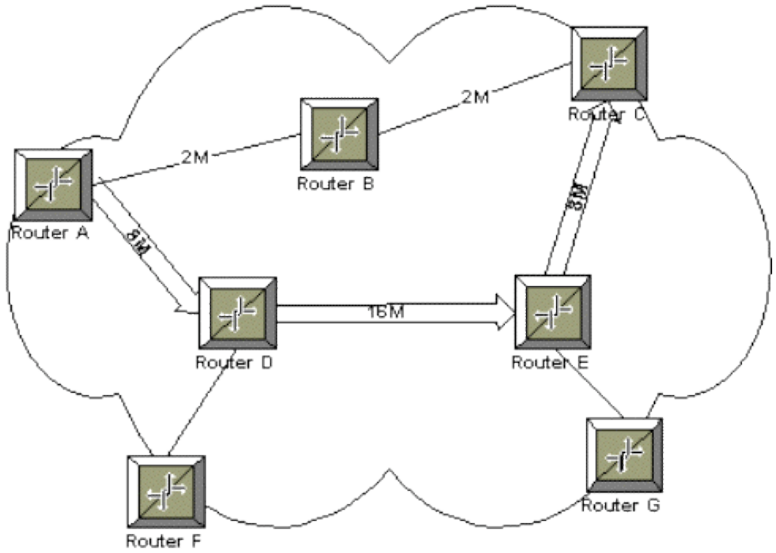
Ejemplo de uso de LFI



QoS routing

QoS Routing

- 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 costos (computacionalmente hablando)
- *Constraint-based Routing*
  - Calcular caminos teniendo en cuenta no solo QoS sino también políticas



$A \rightarrow C$: 4M

Arquitecturas

Propuestas del IETF

- **IntServ** (Integrated Services)
  - Filosofía: reserva de recursos
  - Cada router del trayecto ha de tomar nota y efectuar la reserva solicitada
- **DiffServ** (Differentiated Services)
  - Filosofía: priorización de tráfico
  - El usuario marca los paquetes con un determinado nivel de prioridad
  - Los routers van agregando las demandas de los usuarios y propagándolas por el trayecto
  - Esto le da al usuario una confianza razonable de conseguir la QoS solicitada
- Pueden coexistir

IntServ: Características

- RFC 1633
- Para cada flujo (puede ser agregado) reserva recursos en todo el camino
- Orientado a conexión
- Requiere un protocolo de señalización que soporten todos los routers
- No requiere modificar los protocolos existentes

IntServ: Servicios

- *Best Effort*
- *Controlled load service*
  - RFC 2211
  - *"commitment … to provide … with service closely equivalent to unloaded best-effort"*
  - Prácticamente sin pérdidas
  - No da garantías estrictas
- *Guaranteed service*
  - RFC 2212
  - *"provides firm (mathematically provable) bounds on end-to-end datagram queueing delays."*
  - Garantías de BW
  - Retardo acotado
  - Sin pérdidas en buffers
  - Garantías estrictas

IntServ: *Flow parametrization*

filterspec (*Filter specification*)

- Determina qué paquetes forman el flujo
- Flujo identificado en base a IPs + puertos
- Separa en diferentes colas

flowspec (*Flow specification*)

- **Tspec** (*Traffic specification*)
  - Descripción del tráfico
  - Parámetros de un *tocken bucket* por el que pasa el tráfico
  - Mean rate, tocken bucket depth, max rate, max packet length
- **Rspec** (*Service Request specification*)
  - Requisitos de QoS impuestos a la red
  - BW, retardo, probabilidad de pérdida

IntServ: *Signaling*

- Requisitos
  - Debe poderse usar en redes IP
    - Emplear tablas de rutas existentes
    - Reaccionar ante cambios de rutas
  - Soportar multicast
    - Flujos que se agregan en árbol
  - Pequeña sobrecarga
    - Pocos mensajes y pequeños
  - Modular y fácil de extender

- Resultado:
  - RSVP (*Resource reSerVation Protocol*)
  - RFC 2205
  - *Soft state (periodic updates)*
  - ¡ No sirve para calcular el camino !
  - Empleado en IntServ, DiffServ, MPLS, …

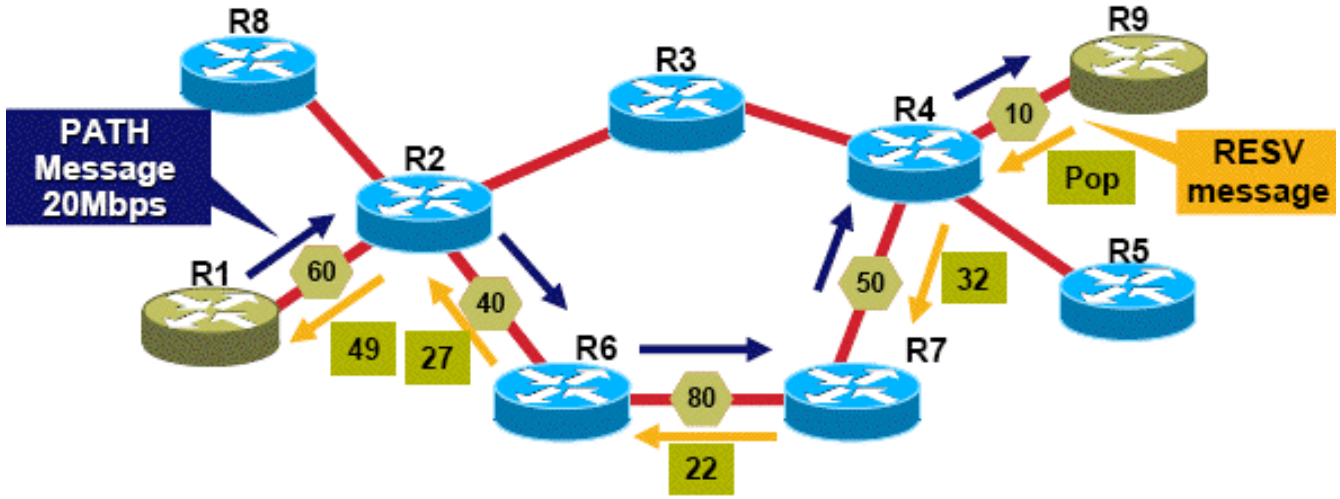
RSVP: Mensajes

PATH

- Desde fuente de tráfico, Tspec
- Establecer camino
- Puede hacerse CAC

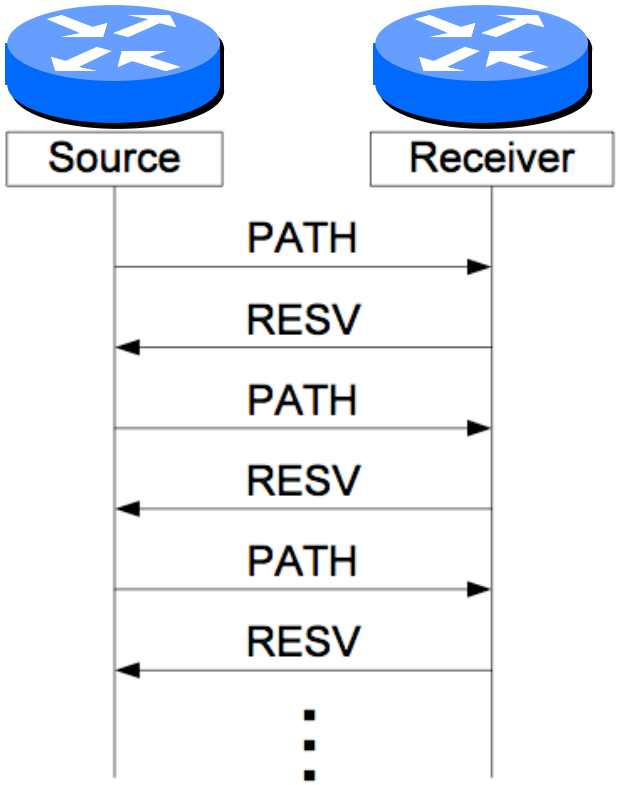
RESV

- Camino inverso al PATH
- Incluye Rspec
- Hacer la reserva en los routers



RSVP: *states*

- Actualizaciones periódicas refrescan el estado
- Se libera al dejar de recibir actualizaciones
- Alternativa (no soportada): Hard state
  - Se mantiene hasta liberarlo explícitamente
  - Requiere algoritmo ante errores

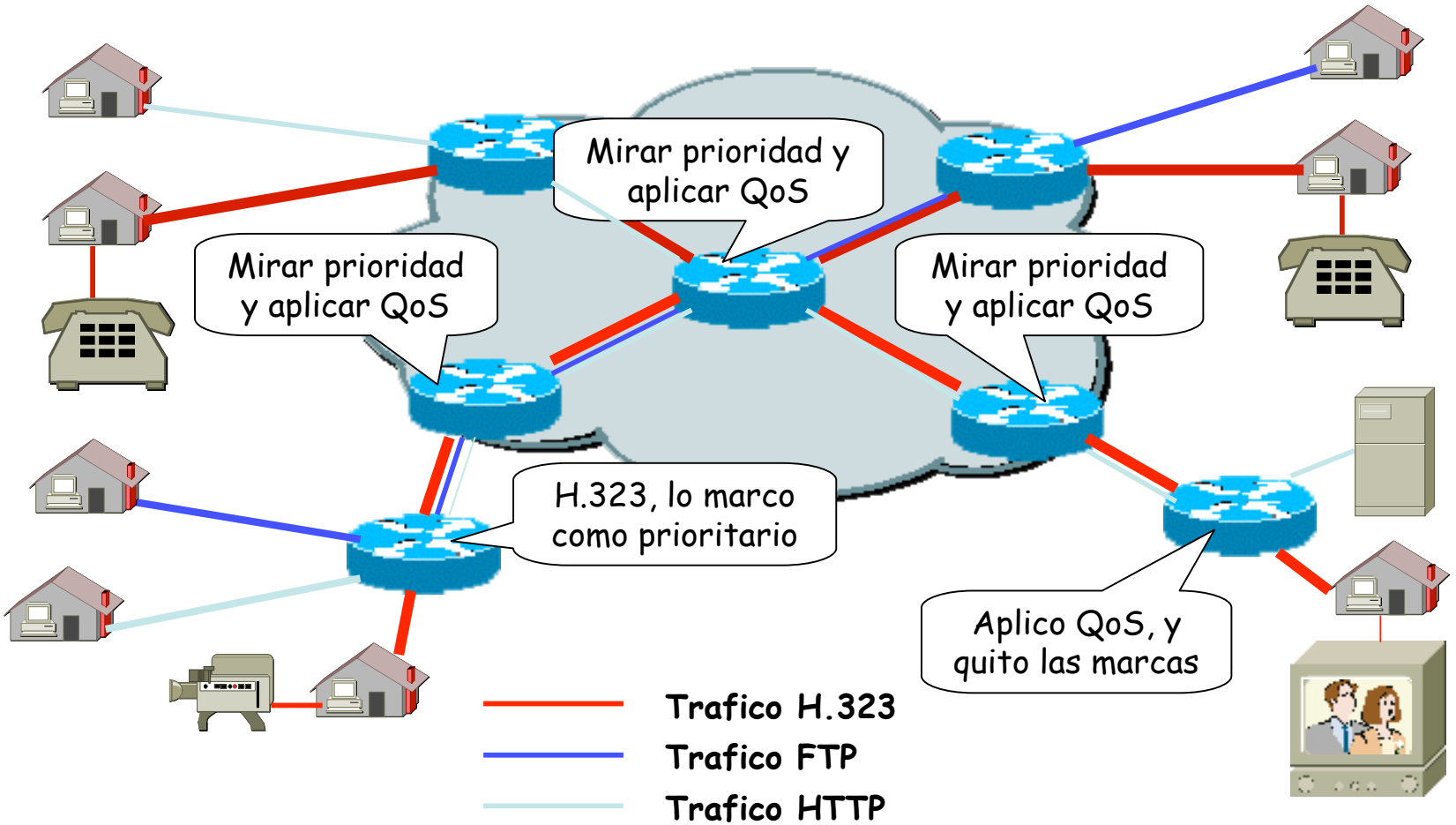


DiffServ

- IntServ no escala bien

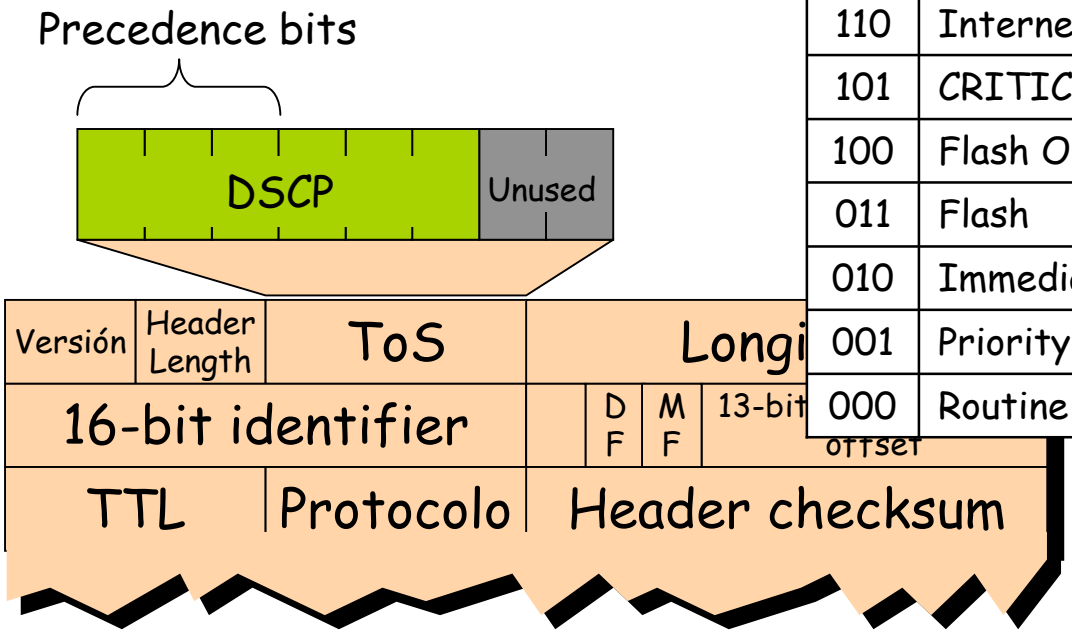
- RFC 2475, 2638
- Clasificar el tráfico en pocas clases
- Clasifican los *ingress routers* (complejidad en la frontera) con un *codepoint* en la cabecera IP
- DiffServ mapea en cada nodo el *codepoint* en el paquete a un PHB en concreto
- PHB = *Per Hop Behavior*
  - El tratamiento que se le da al paquete 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

Ejemplo


Mirar prioridad y aplicar QoS
Mirar prioridad y aplicar QoS
Mirar prioridad y aplicar QoS
H.323, lo marco como prioritario
Aplico QoS, y quito las marcas
Trafico H.323
Trafico FTP
Trafico HTTP

DiffServ: DSCP

- Originalmente 3 primeros bits del ToS = *Precedence* bits
- Fácil mapeo a 802.1p bits
- ToS ahora se llama DS (*Differentiated Services*)
- 6 de sus bits son el DSCP (*Differentiated Services CodePoint*)
- *Class Selector Codepoint:*
  - CSx = XXX000
  - Compatibilidad con *precedence*



CSx	Significado histórico	Uso generalizado
111	Network Control	Tráfico de control (ej: routing)
110	Internetwork Control	
101	CRITIC/ECP	Voz
100	Flash Override	Vconf., streaming
011	Flash	Call signaling
010	Immediate	Libres para clasificar tráfico de datos
001	Priority	
000	Routine	*default*

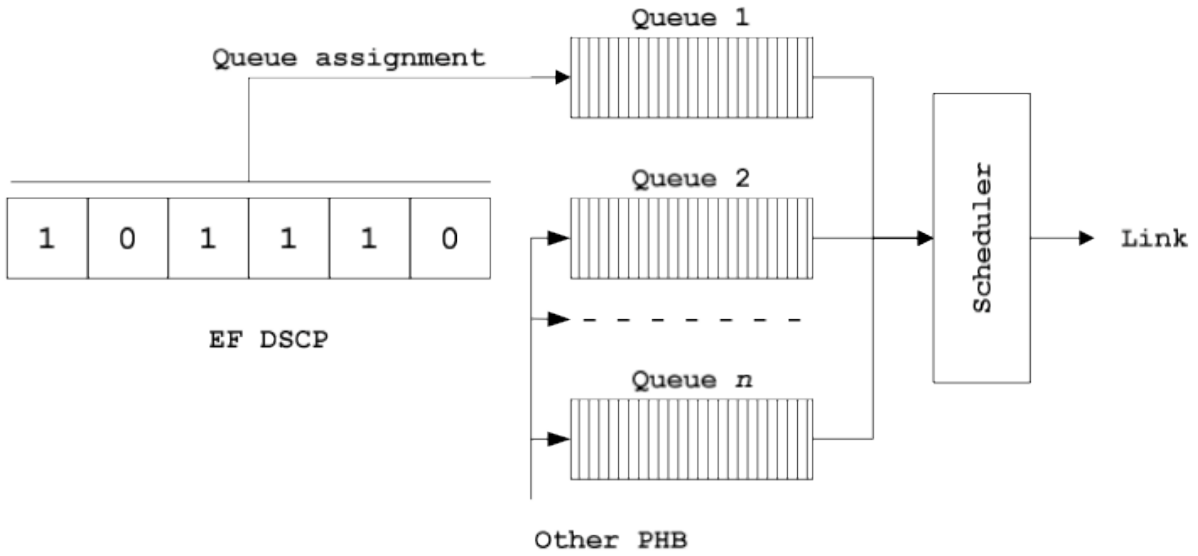
PHBs

PHBs

- *Best-Effort* (BE)
- *Assured Forwarding* (AFxy)
- *Expedited Forwarding* (EF)

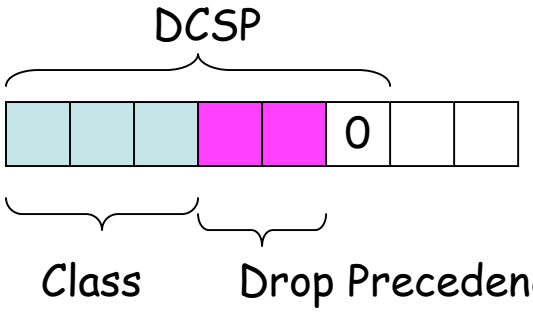
PHB: Expedited Forwarding (EF)

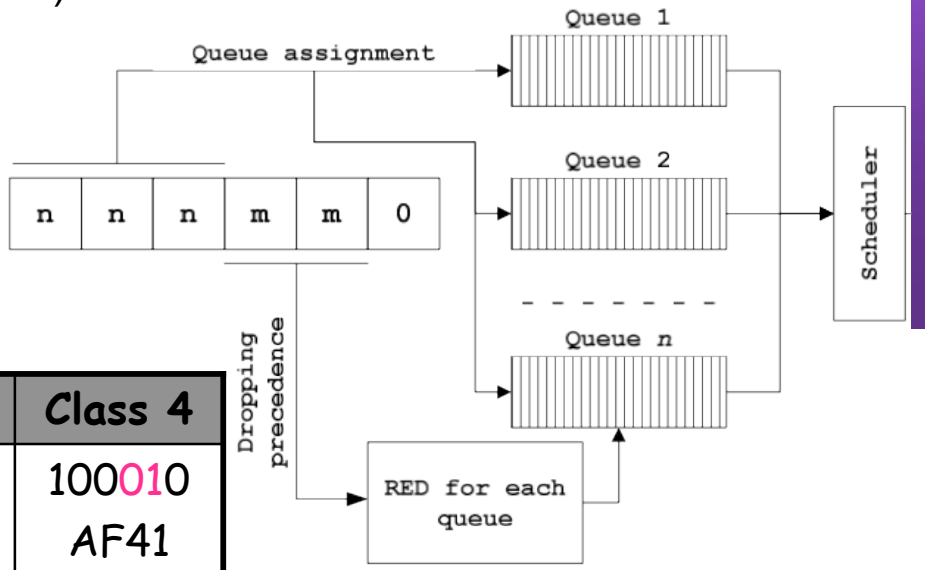
- RFC 2598
- Alta prioridad
- DSCP 10110
- *"…the departure rate of the aggregate's packets from any diffserv node must equal or exceed a configurable rate."*
- *"The EF traffic SHOULD receive this rate independent of the intensity of any other traffic attempting to transit the node. "*
- Este PHB se puede implementar con PQ, WRR, CBQ, etc.



PHB: Assured Forwarding (AF)

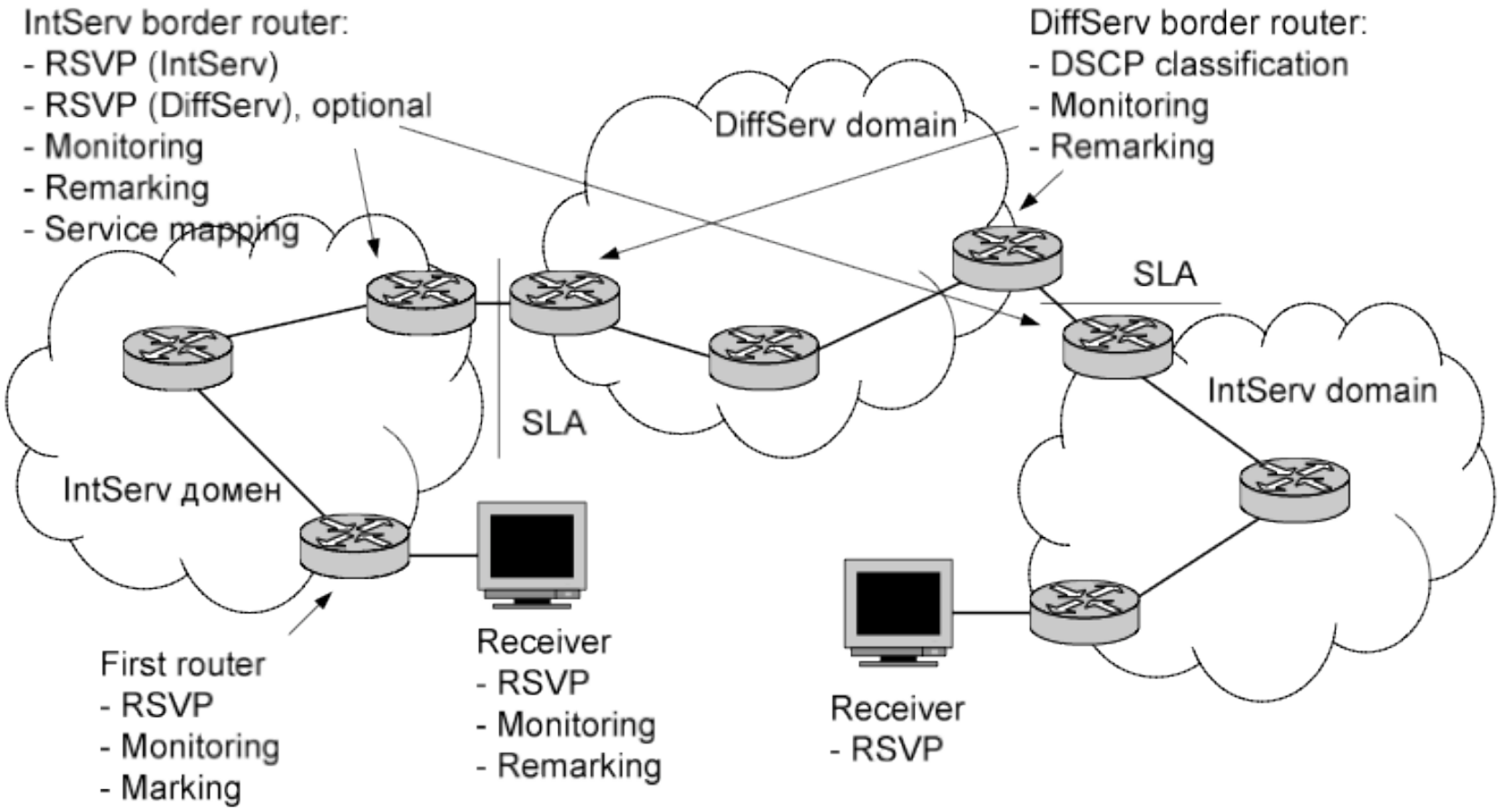
- RFC 2597
- Se definen 4 clases (AF1x, AF2x, AF3 y AF4x)
- Cada una tiene una reserva en cada nodo (BW, buffer)
- Cada una con 3 probabilidades de descarte (*drop*)
- DSCP xxxyy0 : xxx la clase, yy la prob. Descarte
- Debe emplear AQM (RED, WRED, …)





Drop	Class 1	Class 2	Class 3	Class 4
Low	001010 AF11	010010 AF21	011010 AF31	100010 AF41
Medium	001100 AF12	010100 AF 22	011100 AF32	100100 AF42
High	001110 AF13	010110 AF23	011110 AF33	100110 AF43

IntServ over DiffServ



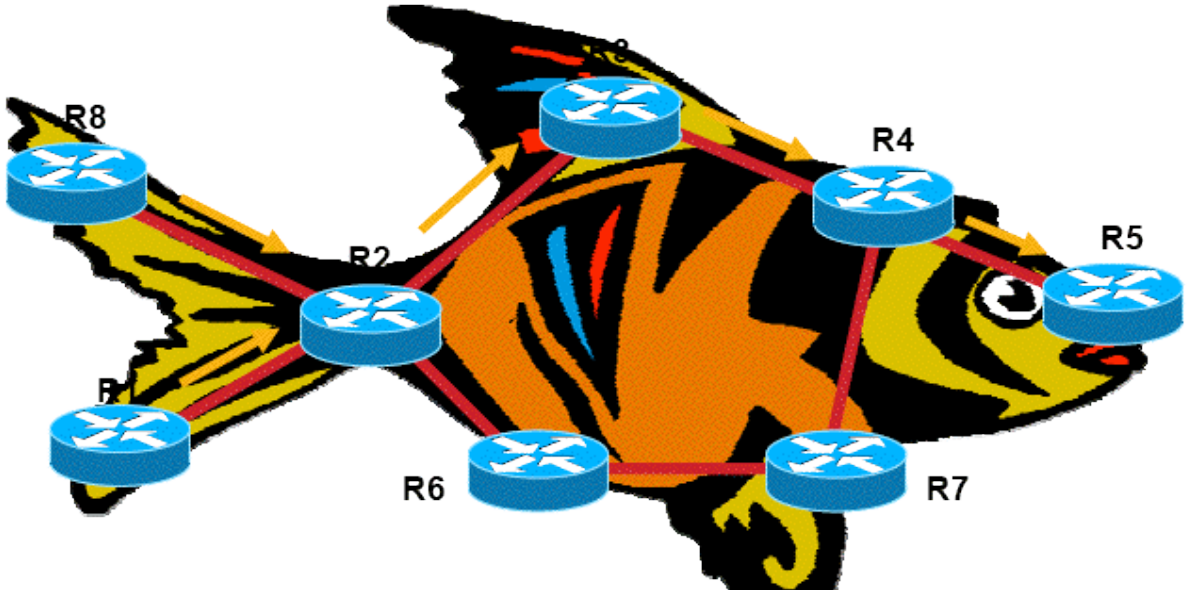
Traffic Engineering

Traffic Engineering (TE)

- RFC 3272 (Overview and Principles of Internet Traffic Engineering)
- *".. that aspect of Internet network engineering dealing with the issue of performance evaluation and performance optimization of operational IP networks."*
- *"[TE] encompasses the application of technology and scientific principles to the measurement, characterization, modeling, and control of Internet traffic."*
- Existe desde las redes telefónicas clásicas
- Proceso:
  - *Measurement*: desde el nivel de paquete al de flujo, usuario, agregado de tráfico o red
  - *Modeling, Analysis and Simulation*
  - *Optimization*: desde real-time optimization a network planning

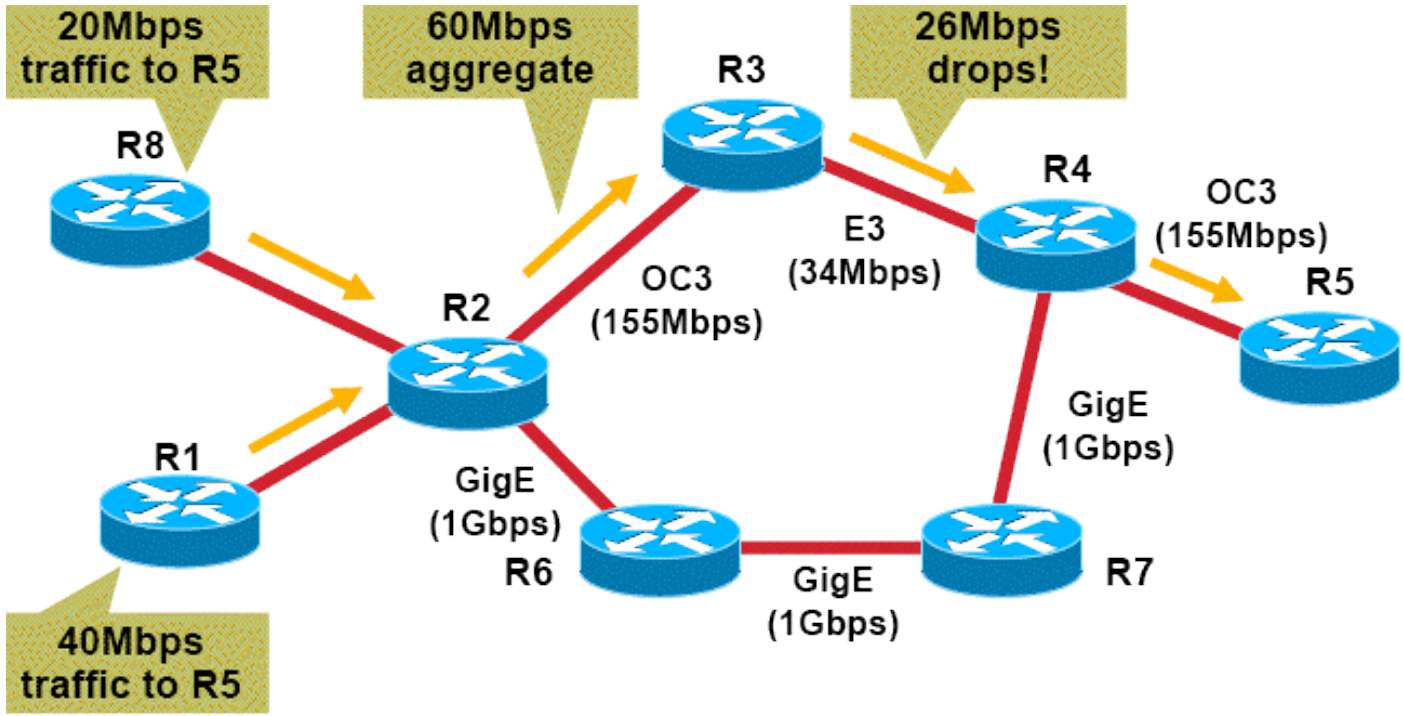
Traffic Engineering

- Network Engineering
  - Construir la red para transportar el tráfico esperado (¡predecir!)
- Traffic Engineering
  - Manipular el tráfico para encajar en la red
  - Prevenir enlaces congestionados y otros infrautilizados
- No podemos contar con predecir los patrones de tráfico
- Seguramente tendremos una red con BW simétricos pero flujos asimétricos
- RFC 2702 - Requirements for Traffic Engineering over MPLS
- Ejemplo: *"The Fish"*



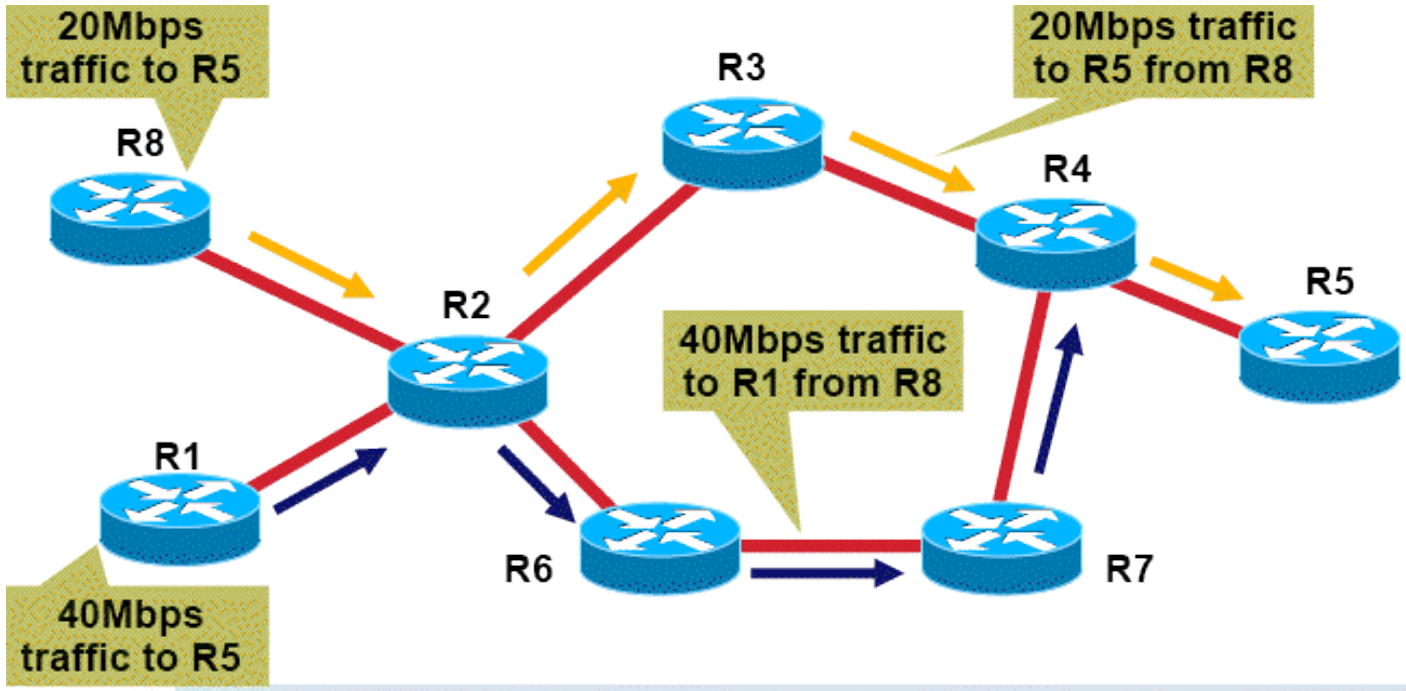
The Fish

- Sin TE



The Fish

- Con TE

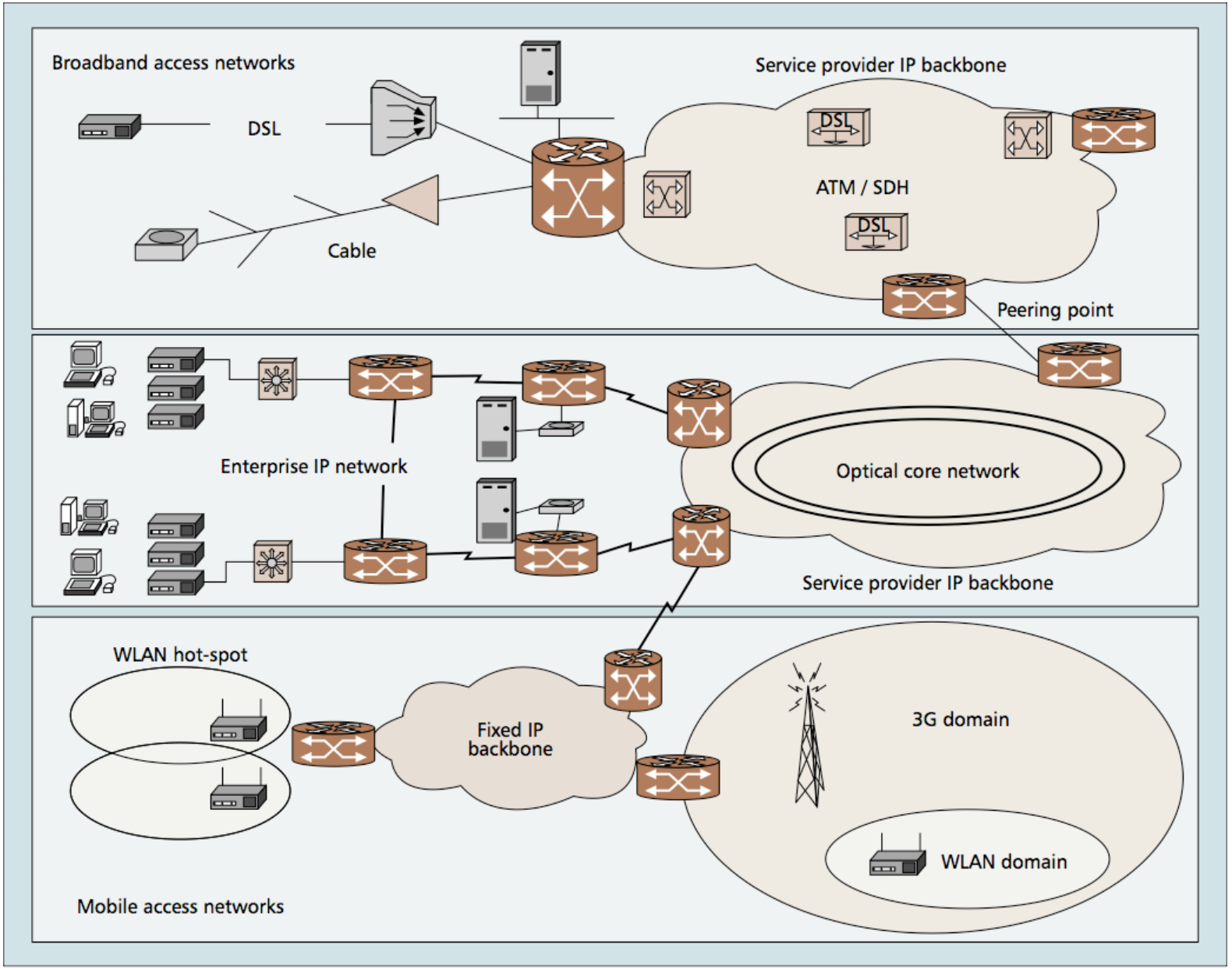


- MPLS Labels can be used to engineer explicit paths
- Tunnels are UNI-DIRECTIONAL

Normal path: R8 → R2 → R3 → R4 → R5

Tunnel path: R1 → R2 → R6 → R7 → R4

Redes heterogéneas



Resumen

- Técnicas:
  - Clasificación y marcado
  - Policing y shaping
  - Scheduling
  - Queue management
  - CAC
  - QoS routing
- Arquitecturas:
  - IntServ
  - DiffServ