QueueManagement + QoS

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

Redes
4º Ingeniería Informática

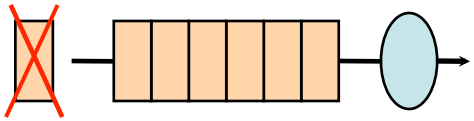
Hoy...

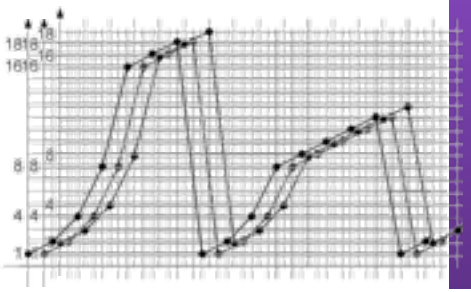
1. Introducción a las redes
2. Tecnologías para redes de área local
3. Conmutación de circuitos
4. Tecnologías para redes de área extensa y última milla
5. Encaminamiento
6. **Arquitectura de conmutadores de paquetes**
   - Scheduling / planificación
   - **Queue Management**
   - **Arquitecturas de QoS**
- Control de acceso al medio
- Transporte extremo a extremo

Queue Management

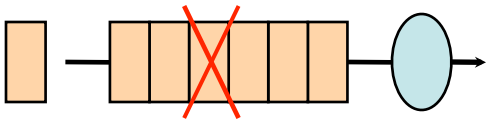
Pasivo

- *Drop-Tail* (la más habitual)
  - Simple
  - Provoca injusticias entre fuentes sincronizadas en epocas de congestión
  - Introduce sincronización global cuando hay varias conexiones TCP atravesando ese enlace
  - Controla la congestión pero no la evita, posible synch.





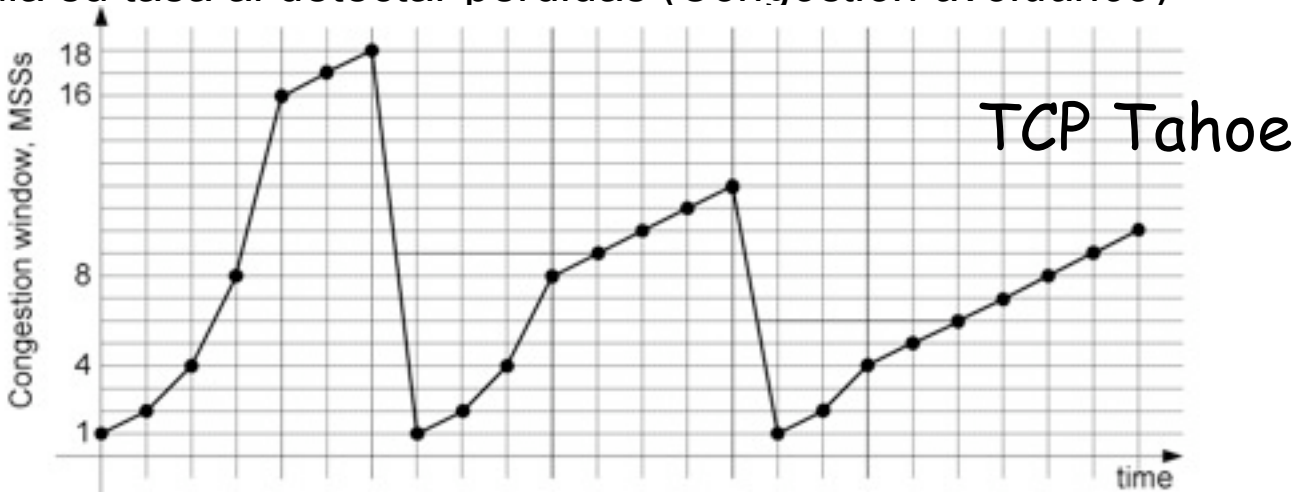
- *Head-drop*
  - Tira los paquetes que más tiempo llevan en el buffer
  - Probablemente ya han sido retransmitidos (TCP)
  - Probablemente ya llegan tarde (UDP/RTP)
  - Controla la congestión pero no la evita, posible synch
- *Random-Drop* (ante cola llena)
  - Se puede reducir la sincronización global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



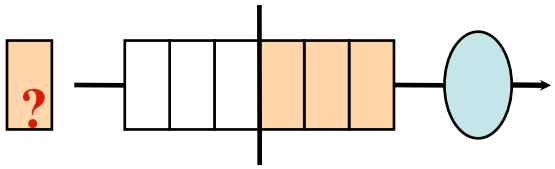
Queue Management

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*)

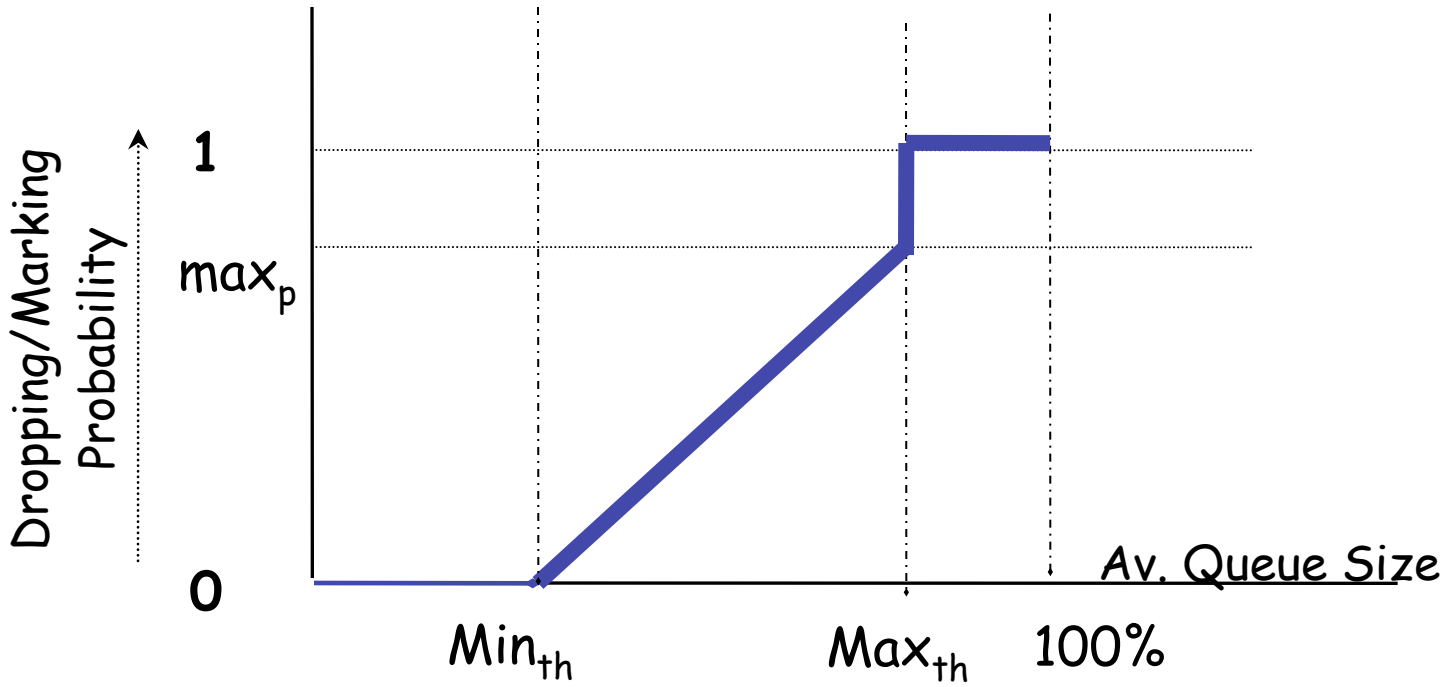


- *Early-Random-Drop* (cola no llena)
  - Si la cola excede un nivel se tira cada paquete que llega con una probabilidad fija



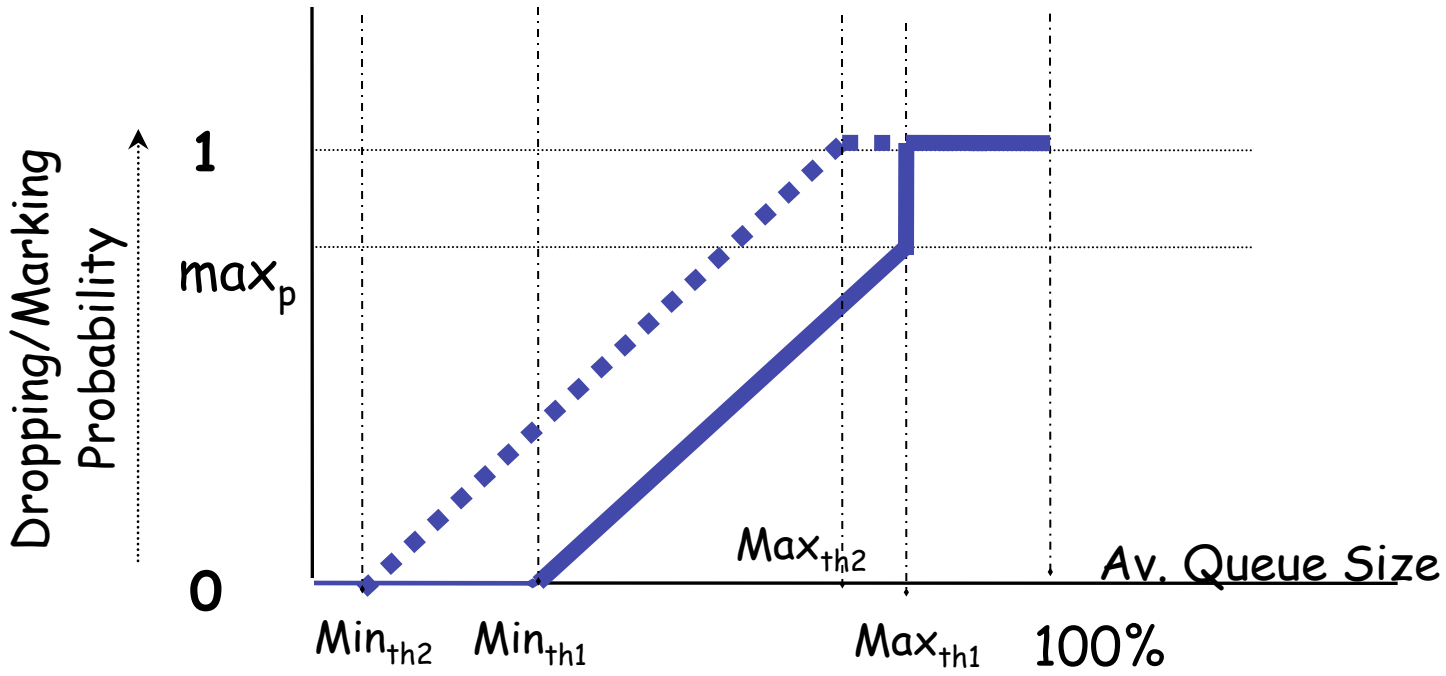
Active Queue Management

- *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



Efectos

- Si el protocolo de transporte reacciona a las perdidas reduciendo su velocidad...

  No solo controlan la congestión sino que la evitan (congestion avoidance)

  Funciona con TCP pero no con UDP

- Los routers con RED ayudan a TCP a reaccionar a la congestión antes de que se produzca

  Ese es otro problema fundamental de redes

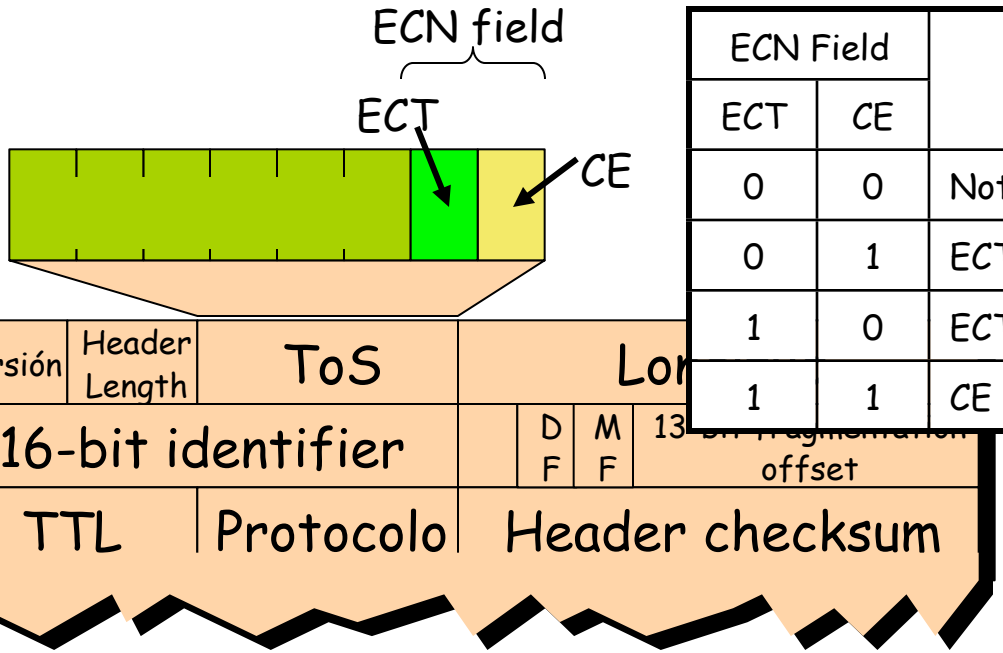
  Detección/evitación de congestión

- Un paso más...

  Detección de congestión asistida por la red

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

En resumen

- Queue management pasivo
  - Drop-Tail, Head-Drop, Random-Drop
- Active queue management (AQM)
  - RED, WRED...
  - Pensado normalmente para TCP

    Protocolo de nivel de transporte que intenta reaccionar a la congestión de la red

  - ECN: notificar al nivel de transporte de que hay congestión

    Problema de la **congestión** en la red

    Mas sobre esto en las clases de TCP

Arquitecturas de QoS

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*)

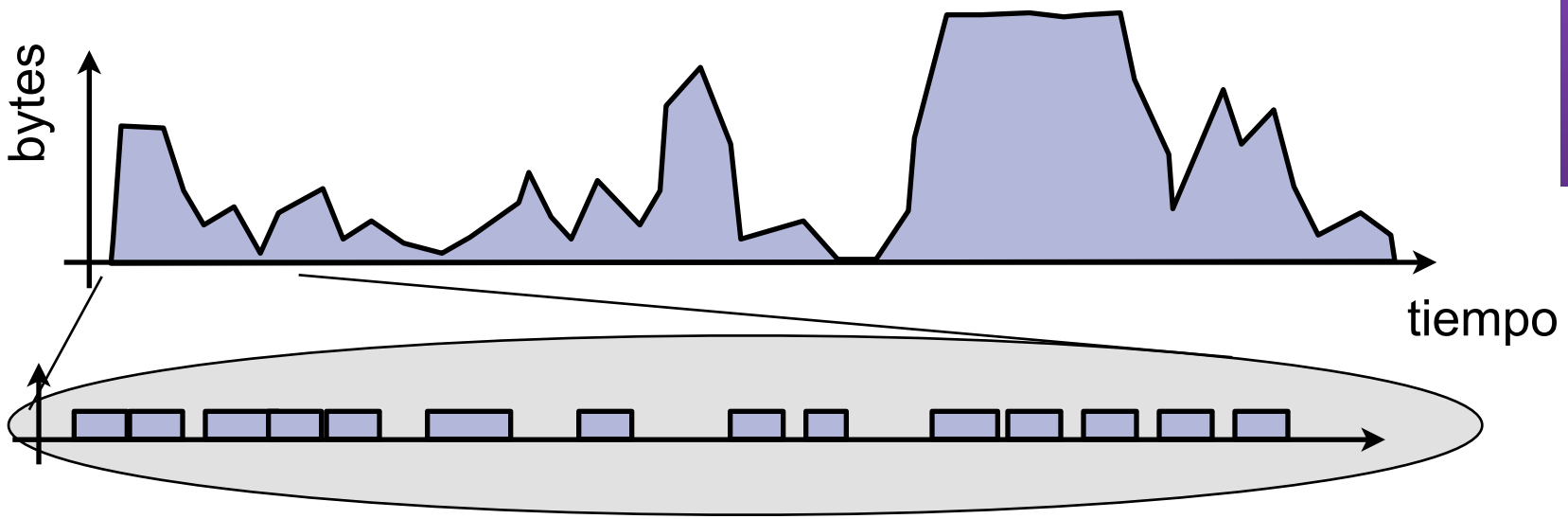
- Determina qué paquetes forman el flujo
- Flujo identificado en base a IPs + puertos
- Separa en diferentes colas

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

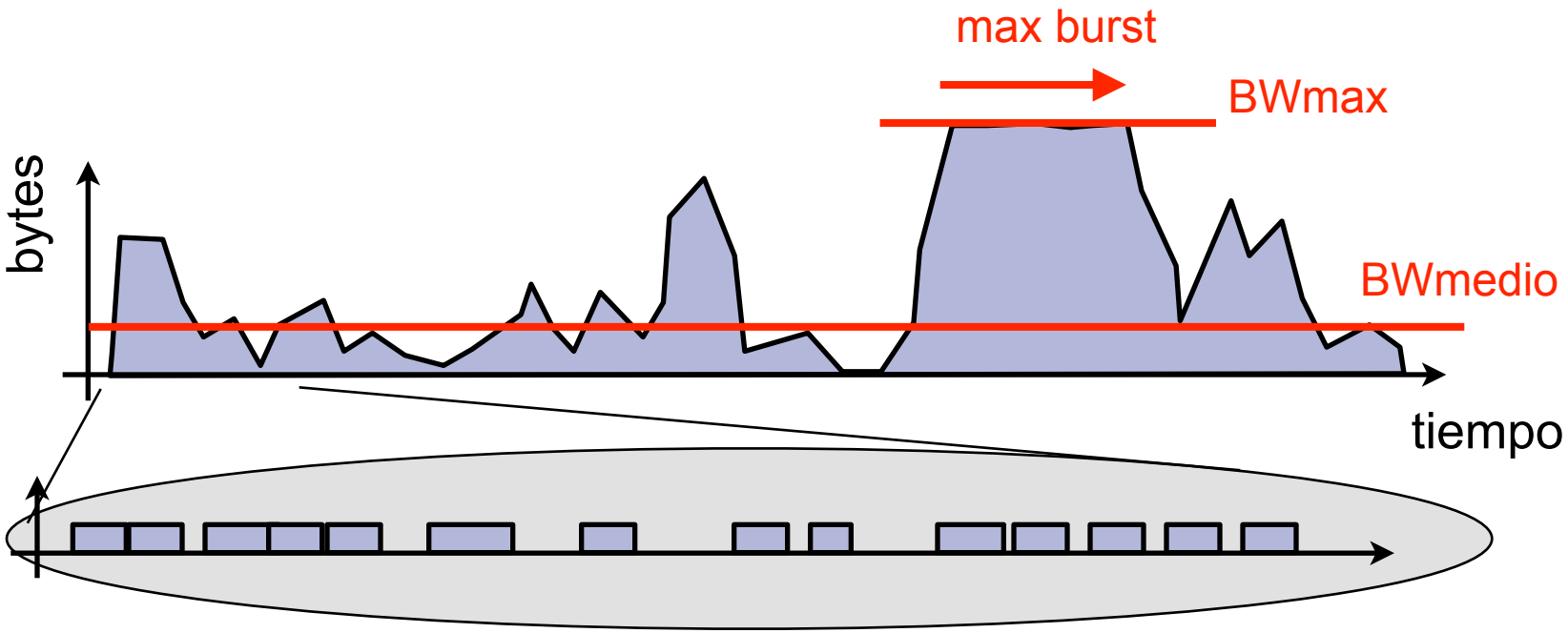
Descripción del tráfico

- Ancho de banda medio (bps)
- Tamaño medio de paquete (bytes)
- Pero normalmente no es regular...



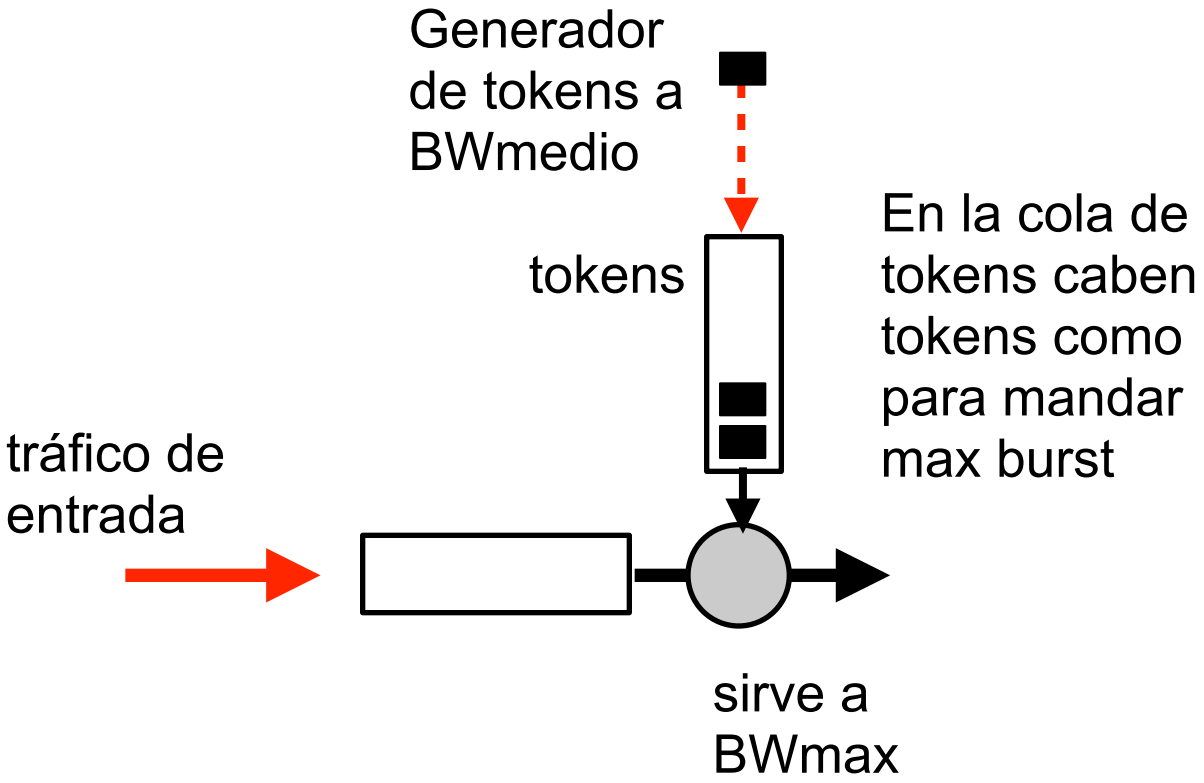
Descripción del tráfico

- Ancho de banda medio (bps)
- Tamaño medio de paquete (bytes)
- Tasa de pico (bps)
- Tamaño máximo de ráfaga sostenida (bytes)



Descripción del tráfico

- Token bucket (leacky bucket)



- El trafico que sale de aqui cumple las limitaciones de BWmax, MaxBurst y BWmedio
- Esto es un regulador (shaper) de tráfico

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 !
  - No es exclusivo de IntServ (MPLS, DiffServ, …)

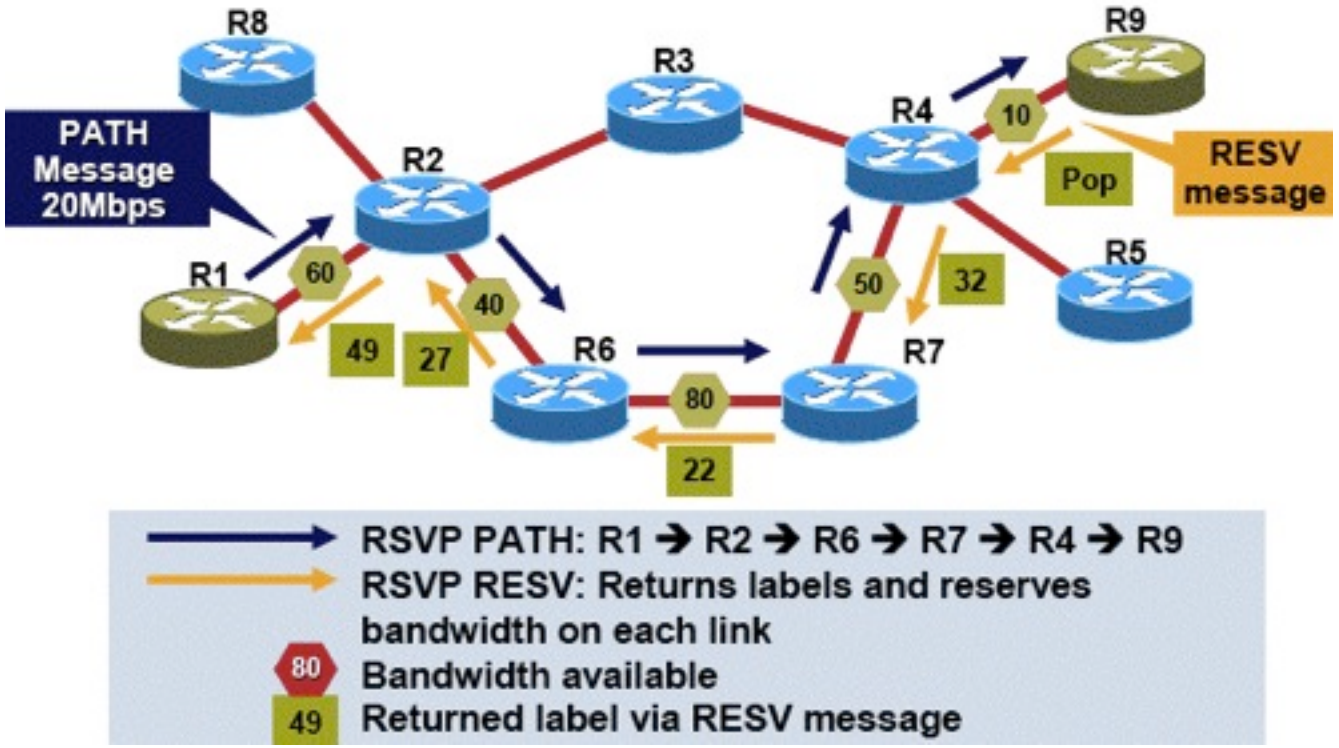
RSVP: Mensajes

PATH

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

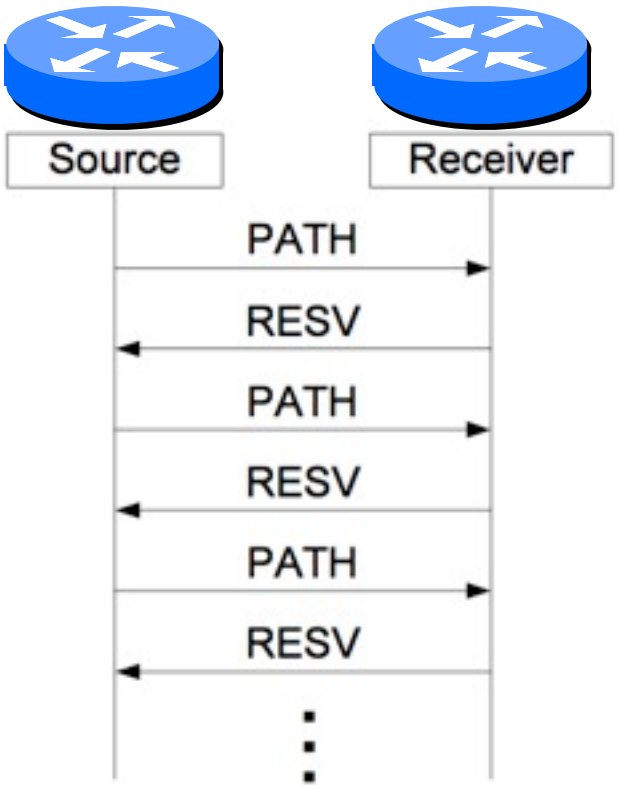
RESV

- 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

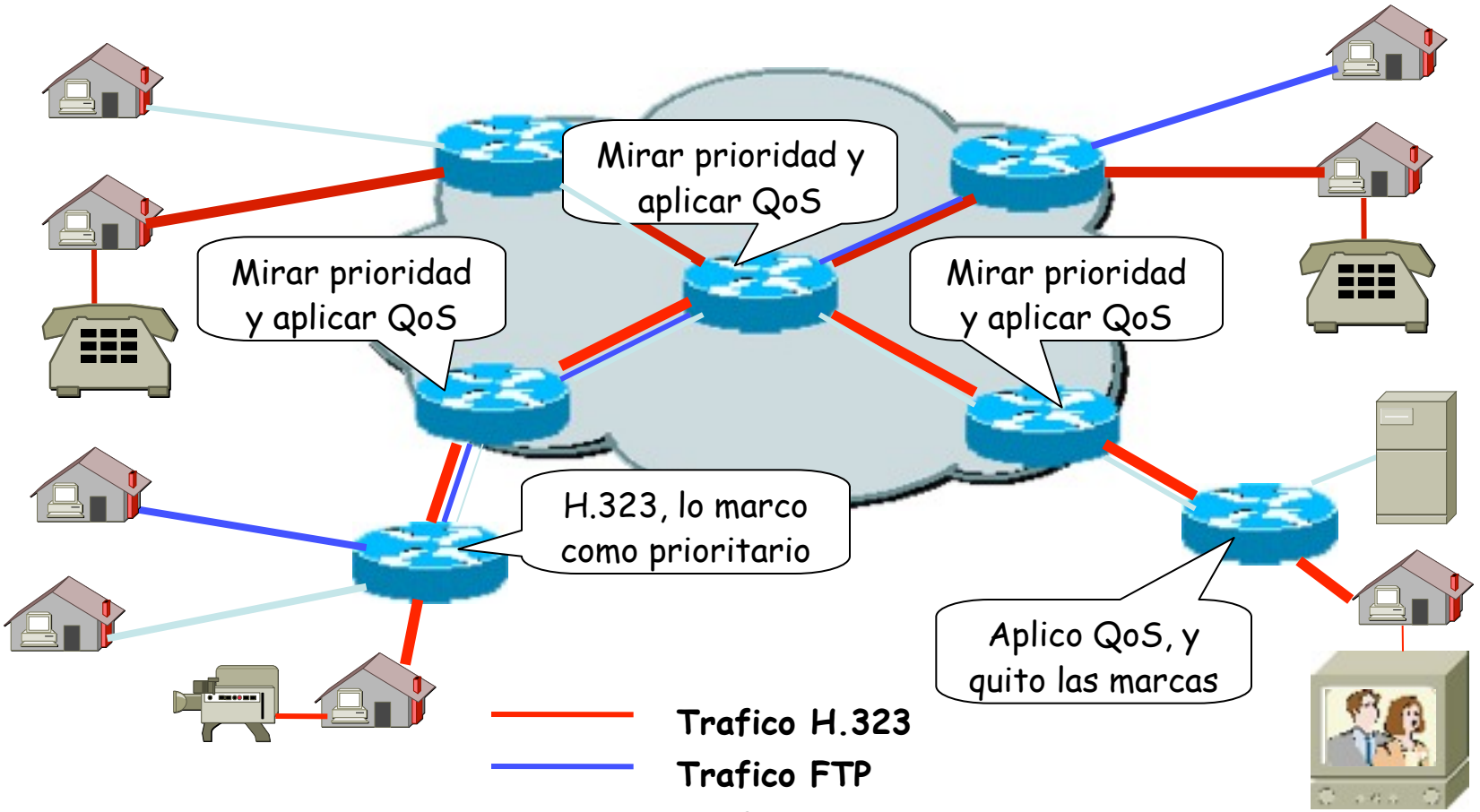


DiffServ

- IntServ no escala bien

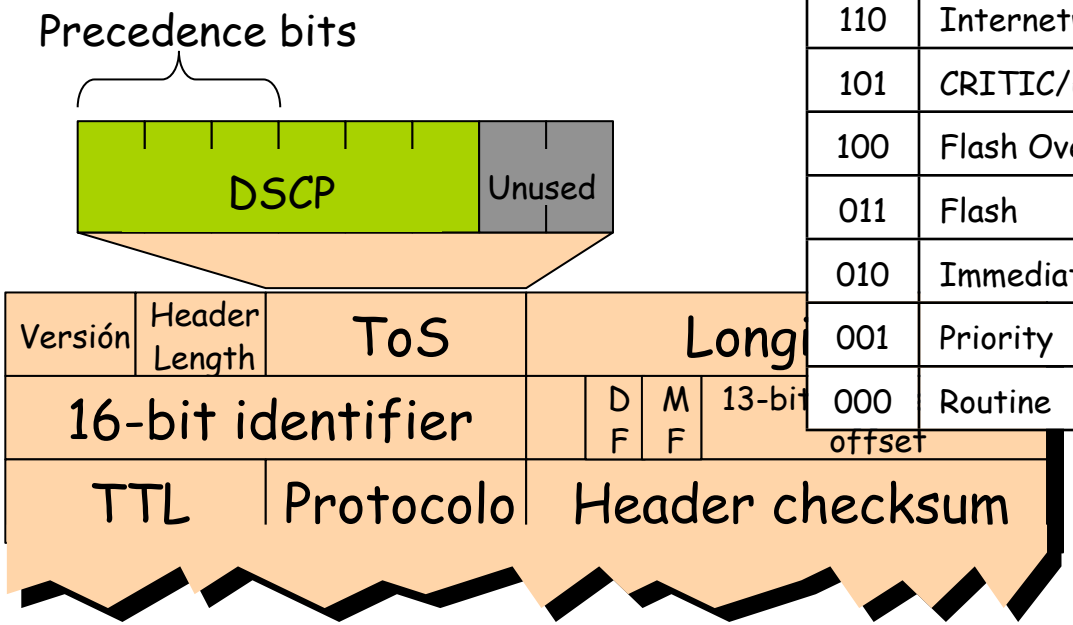
- RFC 2475, 2638
- Clasificar el tráfico en pocas clases
- Clasifican los *ingress routers* (complejidad en la frontera) con un *codepoint* en la cabecera IP
- DiffServ mapea en cada nodo el *codepoint* en el paquete a un PHB en concreto
- PHB = *Per Hop Behavior*
  - El tratamiento que se le da al paquete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

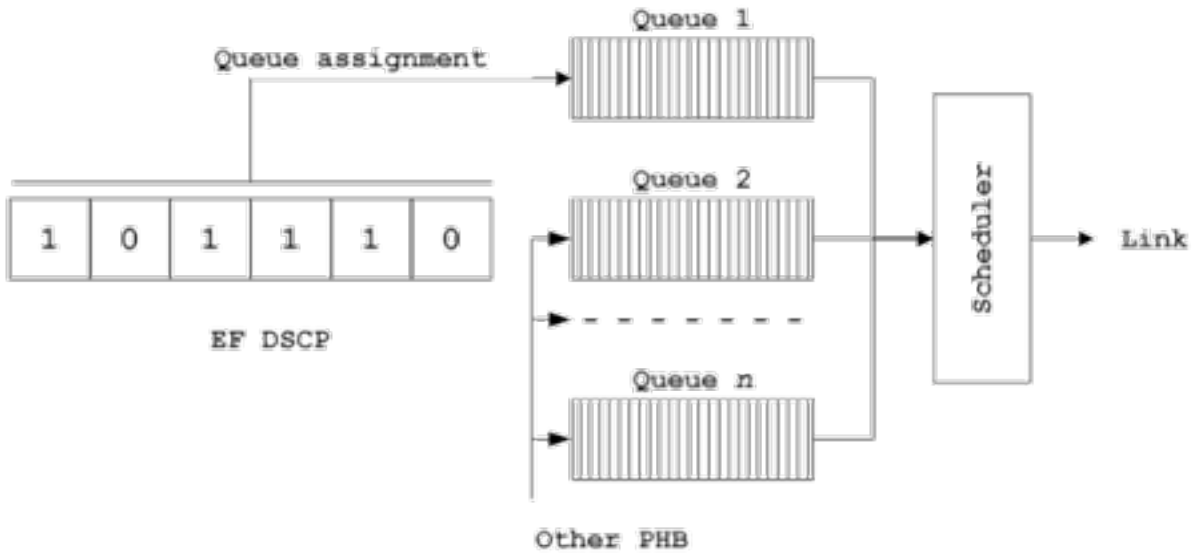
PHBs

PHBs

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

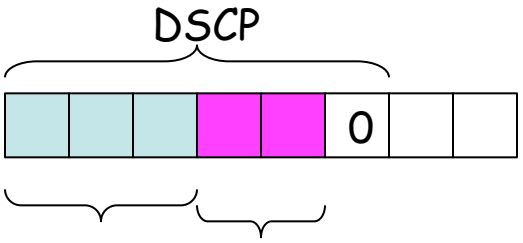
PHB: Expedited Forwarding (EF)

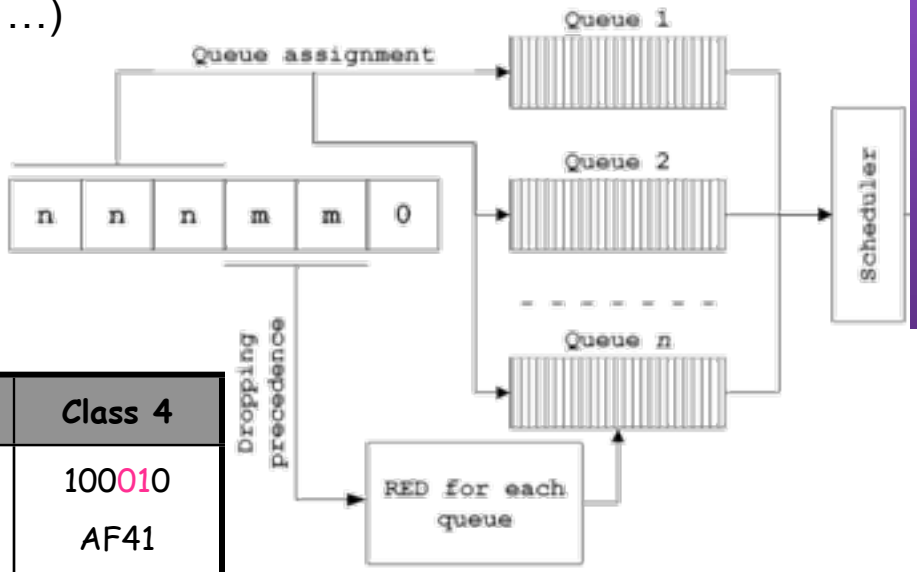
- RFC 2598
- Alta prioridad
- DSCP 10110
- *"…the departure rate of the aggregate's packets from any diffserv node must equal or exceed a configurable rate."*
- *"The EF traffic SHOULD receive this rate independent of the intensity of any other traffic attempting to transit the node. "*
- Este PHB se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

Conclusiones

- Dos arquitecturas para QoS del IETF
  - **IntServ**

    Reserva de recursos

    Establecimiento de sesiones ligeras con RSVP

    Garantias por flujo

    Poco escalable y poco usado
  - **DiffServ**

    Clasificación de tráfico y marcado con el campo DSCP

    Más escalable