

# QoS: Queueing

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

Grado en Ingeniería en Tecnologías de  
Telecomunicación, 3º

# Temas de teoría

1. Introducción
2. QoS
3. Encaminamiento dinámico en redes IP
4. Tecnologías móviles
5. Otros temas

# Objetivos

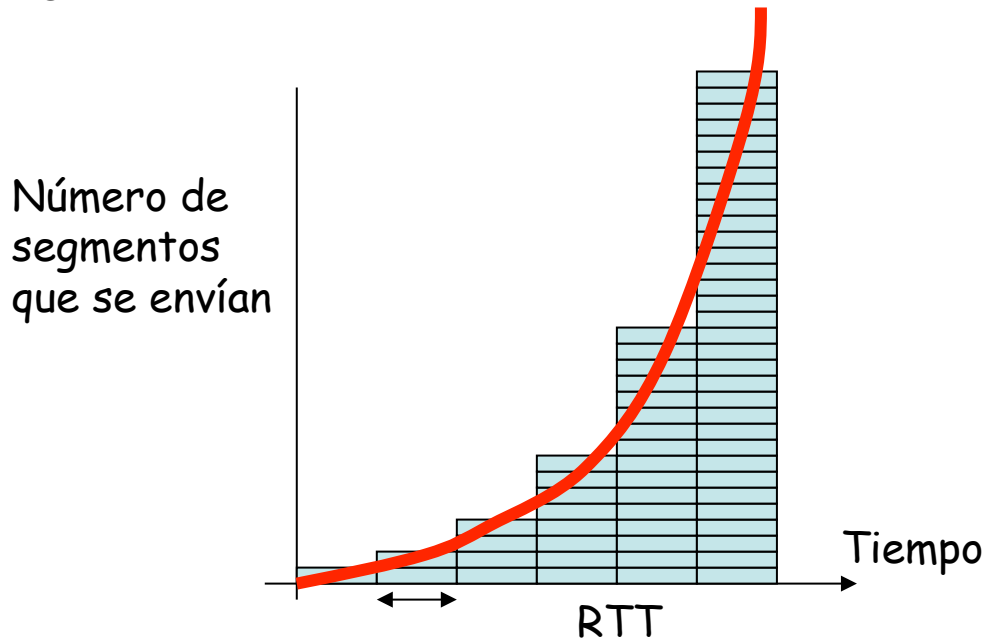
- Conocer los mecanismos de gestión activa de cola

# Congestion Avoidance

- Vamos a ver lo que se llama *Active Queue Management*
- Su objetivo es evitar que interfaces o colas se congestionen
- Diseñado para TCP pues es un protocolo que reacciona ante congestión reduciendo la tasa de envío

# Slow Start

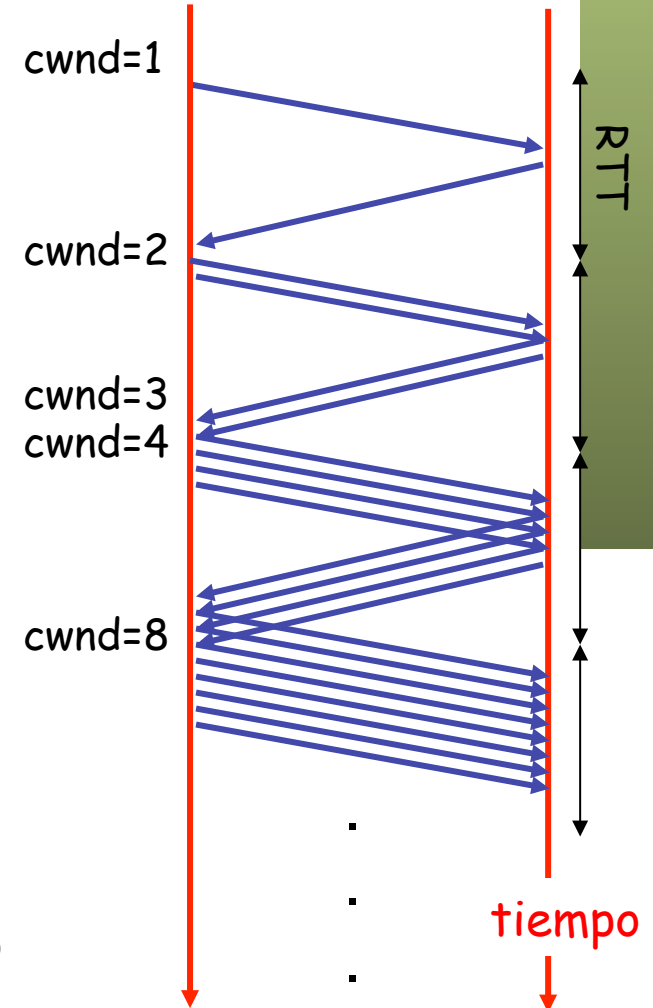
- Al comenzar  $cwnd = 1$  segmento del MSS
  - La ventana es  $\min\{win, cwnd\}$
  - Al principio solo se puede enviar un segmento
  - En realidad (RFC 2581) debe ser:  $cwnd_{inicial} < 2 \times MSS$
- Por cada ACK (que confirme nuevos datos) que se reciba se incrementa  $cwnd$  en 1 segmento del MSS (...)



Emisor



Receptor



# Congestion avoidance

**TCP Tahoe**

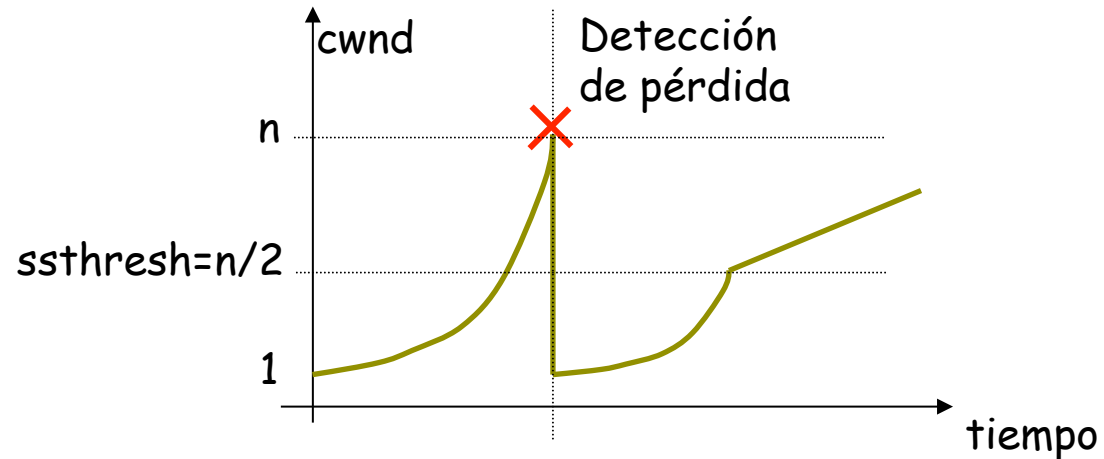
- Inicialización:  $cwnd=1$ ,  $ssthresh=65535$
- *Slow Start* (...)
- Si se detecta una pérdida (...):
  - $ssthresh = \min\{cwnd, win\}/2$   
 [RFC2581]  $ssthresh = \max\{FlightSize/2, 2\}$  (...)

Por **Timeout**:

- $cwnd=1$ , comienza *Slow-start* de nuevo (...)

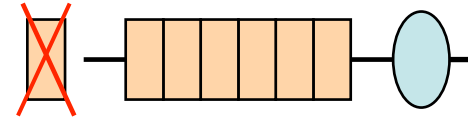
Nuevos datos son confirmados:

- Si  $cwnd \leq ssthresh$  : *Slow-start*,  $cwnd+=1$  (crecimiento exp.) (...)
- Si  $cwnd > ssthresh$  : **Congest. avoidance**,  $cwnd+=1/cwnd$  (crecimiento lineal) (...)

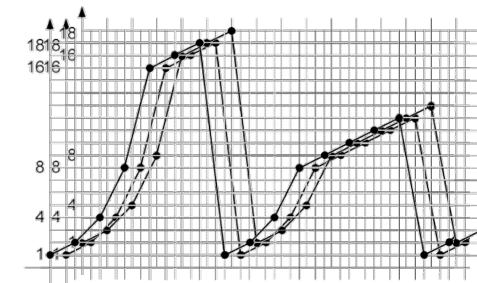


# Passive Queue Management

## Drop-Tail (la más habitual)



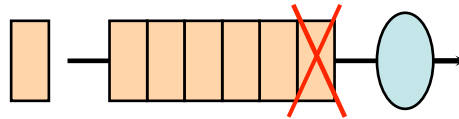
- Simple
- Descarta un paquete independientemente de su importancia
- Controla la congestión pero no la evita
- TCP puede enviar ráfagas (limitadas por el tamaño de la ventana)
- Si llega una ráfaga a una cola casi llena se perderán varios paquetes
- Ante pérdidas TCP reduce ventana de congestión y así velocidad de envío
- Varias conexiones pueden entrar simultáneamente en este proceso de control de la congestión y reducirse en gran medida el throughput global
- Introduce **sincronización global** con varias conexiones TCP en el enlace
- Unos pocos flujos pueden monopolizar el recurso
- La cola se mantiene llena por largos periodos
- Pero la cola es para absorber ráfagas, luego no se podrán absorber
- Colas llenas no lleva a mayor throughput sino a menor throughput
- **Colas poco ocupadas llevan a mayor throughput y menor retardo**



# Passive Queue Management

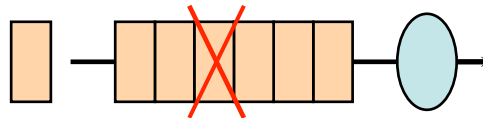
## *Head-drop*

- 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 sincronización



## *Random-Drop (ante cola llena)*

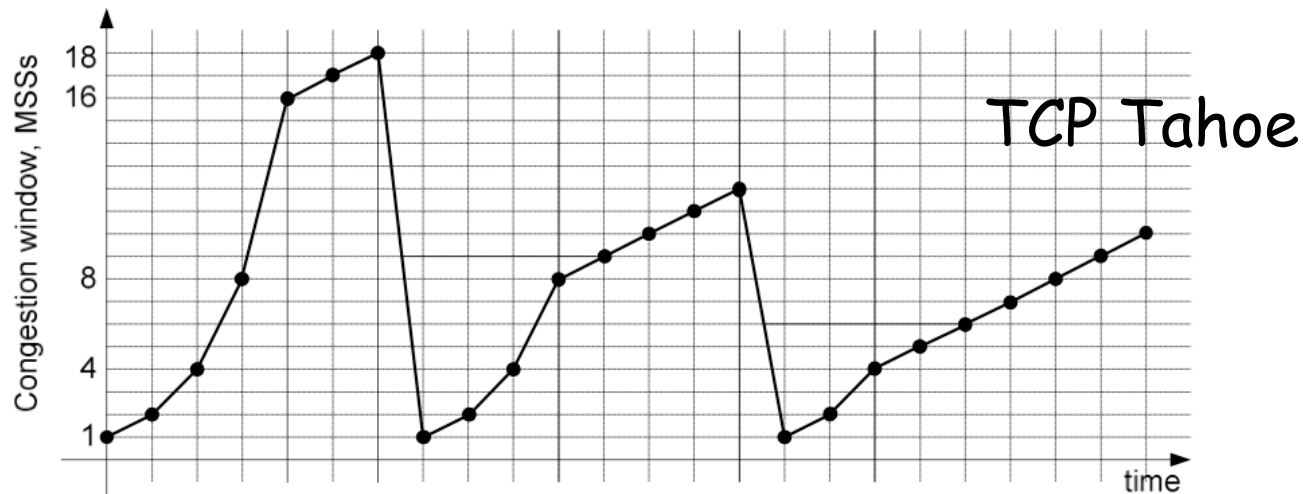
- Se puede reducir la sincronización global pero no controlar UDP
- Controla la congestión pero no la evita





# Active Queue Management

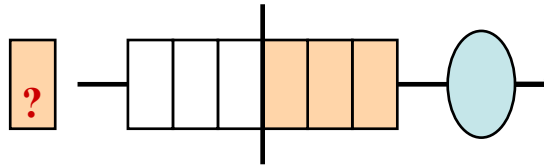
- Pensando en TCP, no controla UDP igual de bien
- Evita sincronizaciones, menores retardos y fluctuaciones
- TCP regula su tasa al detectar pérdidas (*Congestion avoidance*)



# Active Queue Management

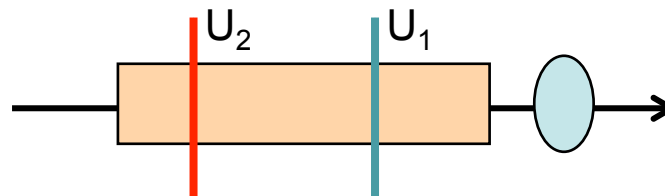
## **Early-Random-Drop** (cola no llena)

- Si la cola excede un nivel se tira cada paquete que llega con una probabilidad fija



## **Weighted Tail Drop**

- Se asignan umbrales en la cola a diferentes clases de tráfico
- Cuando se alcanza el umbral  $U_1$  se descartan los paquetes que lleguen de la clase 1
- Cuando se alcanza el umbral  $U_2$  se descartan también los paquetes que lleguen de la clase 2
- De la tercera clase se descartarán solo cuando se llene la cola

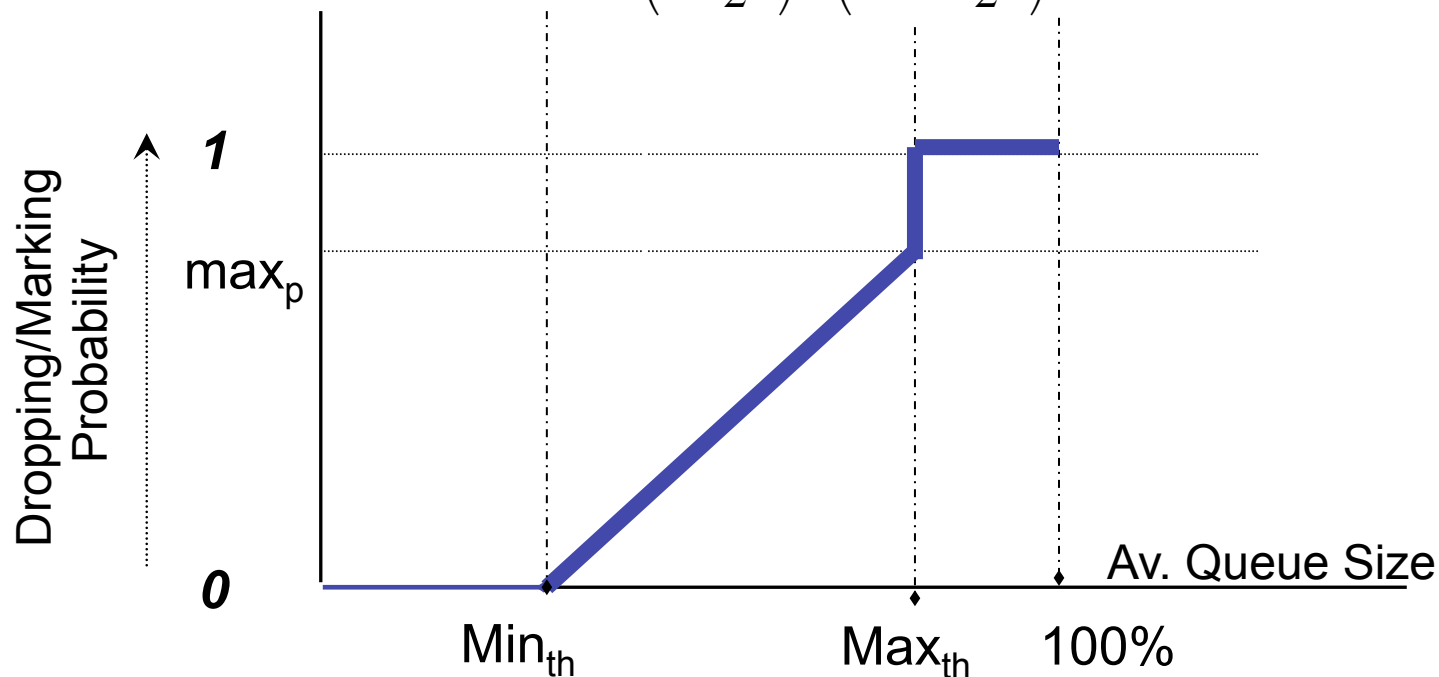


# Active Queue Management

## RED (Random Early Detection)

- RFC 2309 “Recommendations on Queue Management and Congestion Avoidance in the Internet”
- Descartar paquetes probabilísticamente antes de la congestión
- Evalúa la ocupación media del buffer
- Cálculo mediante *exponential moving average*
- Para parámetro  $w$  bajo la media sigue los cambios rápidos del valor instantáneo

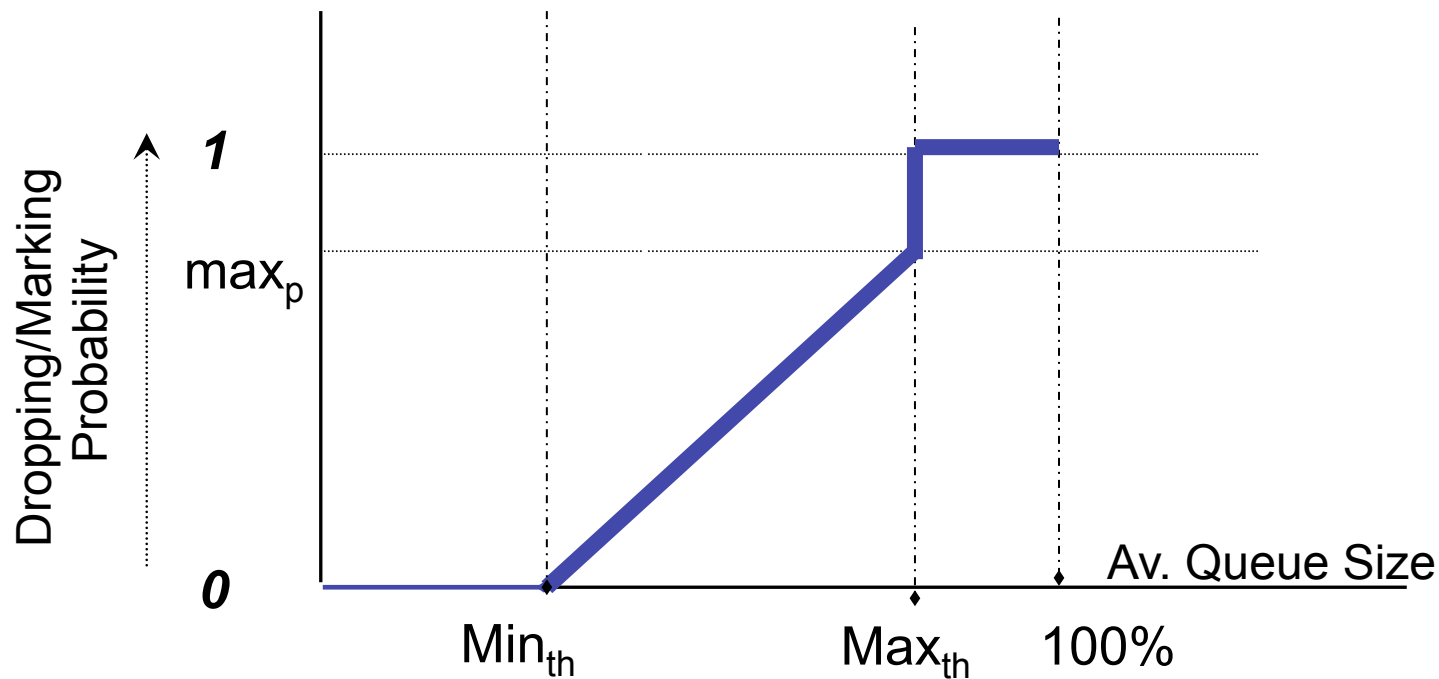
$$q_{avg} = q_{avg\_old} \left(1 - \frac{1}{2^w}\right) + \left(q_{current} \frac{1}{2^w}\right)$$



# Active Queue Management

## RED (Random Early Detection)

- Difícil medir sus beneficios
- Con mala configuración podría comportarse peor que *drop-tail*
- Se han propuesto bastantes más algoritmos de AQM
- RED es el más extendido en implementaciones

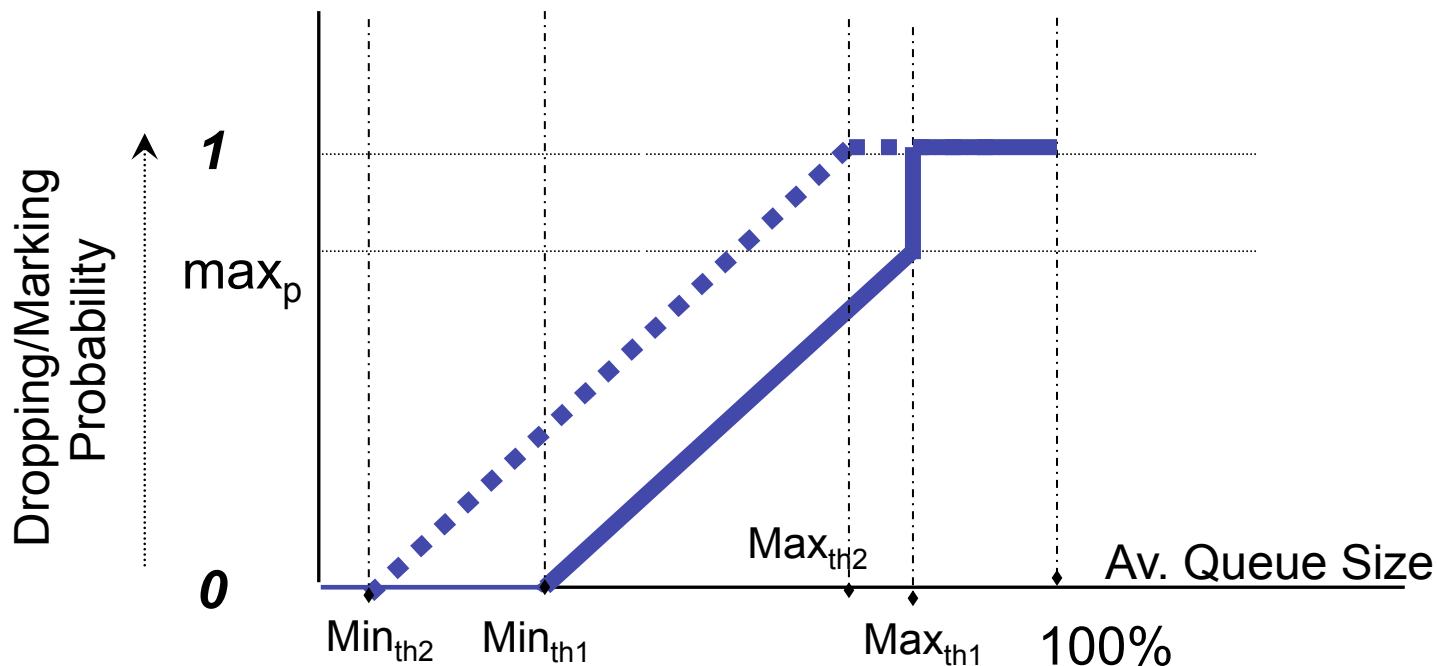


# Active Queue Management

## **WRED (Weighted RED)**

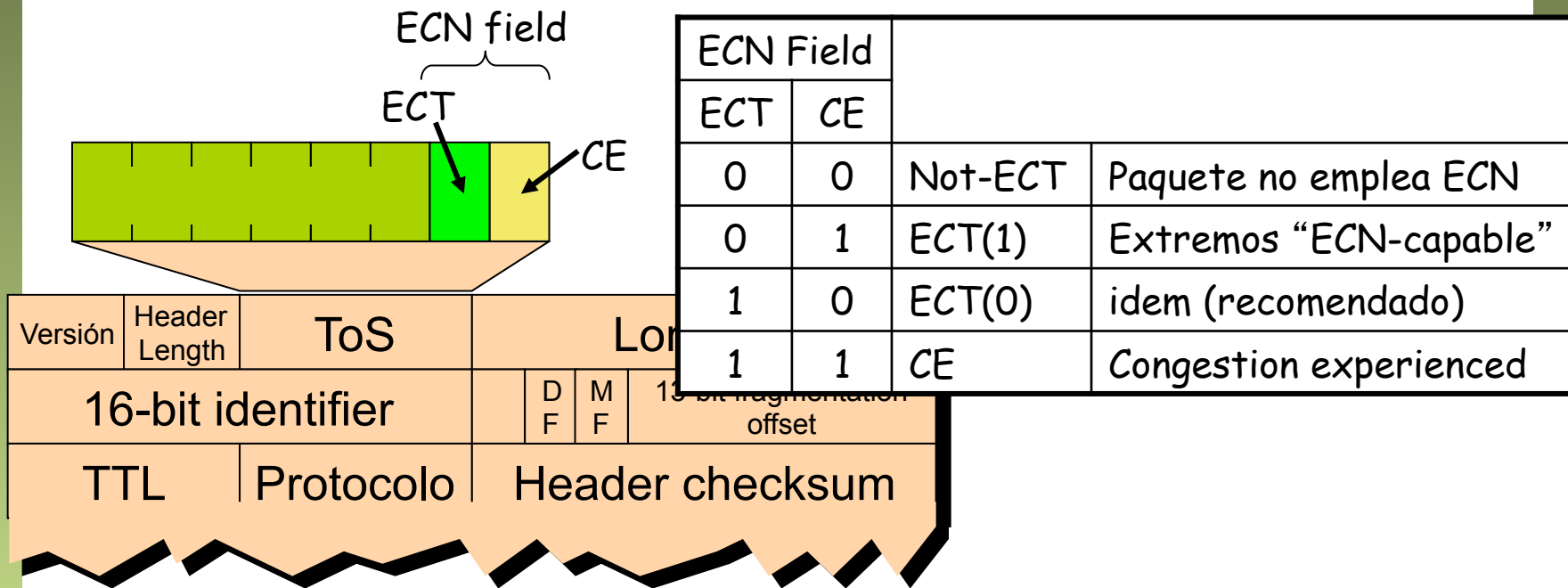
- Emplea un  $\text{Min}_{th}$  y  $\text{Max}_{th}$  diferente para diferentes clases de tráfico
- Mayor cuanto mayor es el valor de precedencia

**Otras propuestas:** Adaptive RED (ARED), RED In & Out (RIO), Flow weighted RED (FRED)...



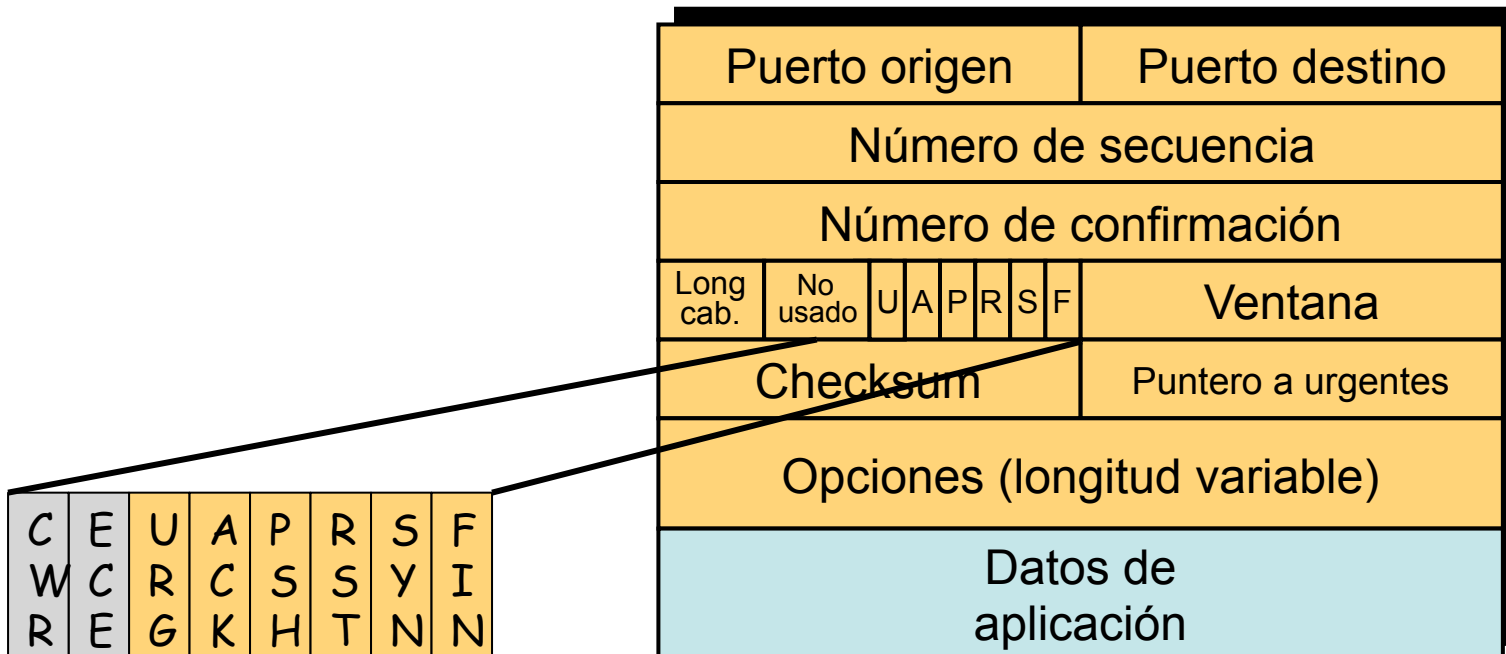
# ECN

- *Explicit Congestion Notification*
- RFC 3168 “The Addition of Explicit Congestion Notification (ECN) to IP
- Extensión a RED: **marcar en vez de descartar** (salvo cuando se alcanza ocupación máxima que sí se descarta)
- Bit ECT = *ECN-Capable Transport*
- Bit CE = *Congestion Experienced*
- Requiere extender el control de congestión de TCP



# ECN y TCP

- Emplea dos flags nuevos en la cabecera TCP
- ECN-Echo (ECE): para que el receptor del paquete con CE activo devuelva esta indicación a emisor
- Emisor reacciona como si hubiera detectado una pérdida
- CWR flag: emisor notifica a receptor en el siguiente paquete de que ha recibido el ECE
- Poco extendida la implementación de ECN en routers y hosts



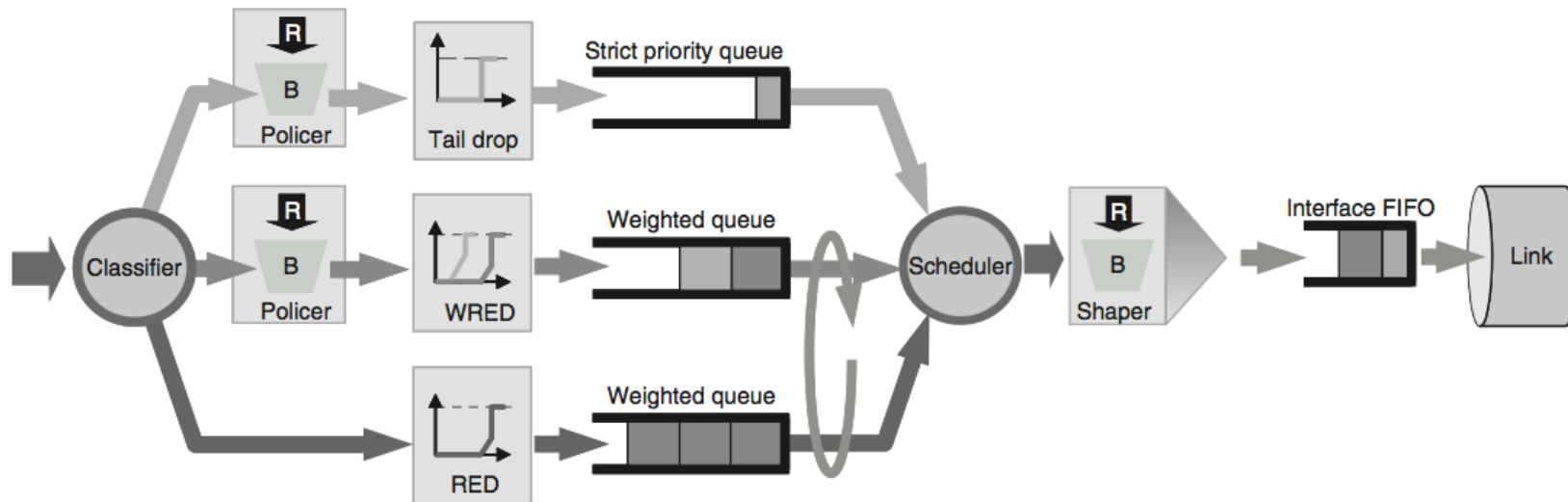


# Implementación

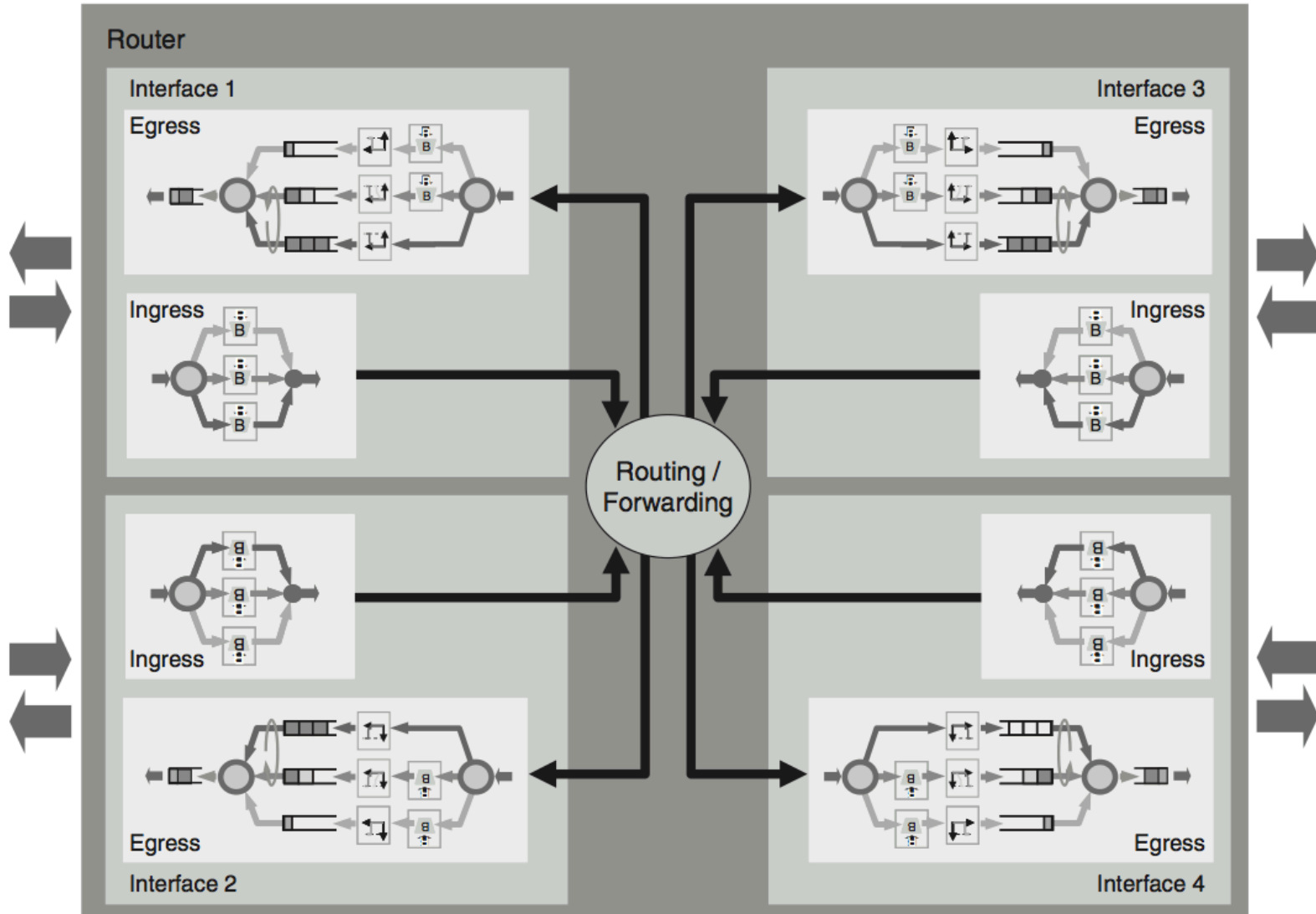




# Implementación



# Implementación



# Resumen

- TCP implementa control de congestión
- Interesa reducir la carga para que las colas no se saturen
- Introducirle pérdidas prematuras a TCP sirve para hacerle regular su tasa
- RED/WRED es la solución más común y flexible pero compleja de configurar