



CEU

## 4 La capa de red II

### 4.1 Calidad de Servicio (1/2)

RdE 2014-2015

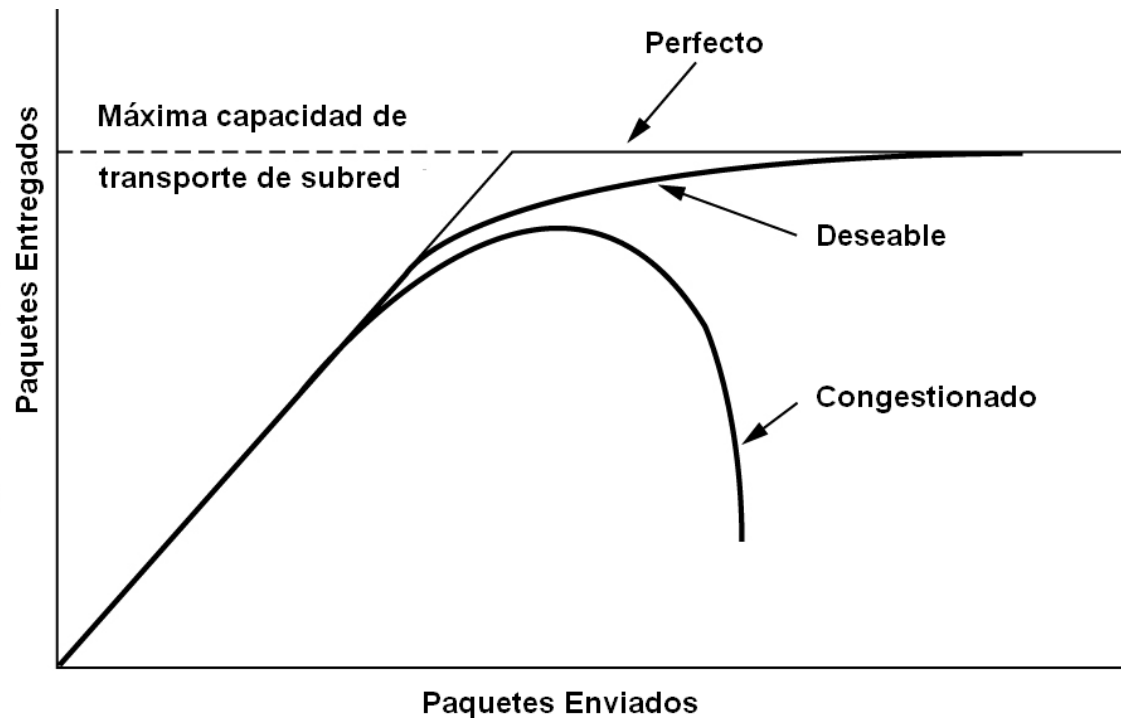
## 4 Guión del Tema 4

- 4 CAPA de RED II:
  - **4.1 Calidad de servicio.**
  - 4.2 Internetworking.
  - 4.3 IPv6.



C

## 4.1 Congestión



- Causas de la congestión:
  - ☐ Línea lenta.
  - ☐ Router lento.
  - ☐ Escasez de memoria.
  - ☐ Demasiada memoria.
  - ☐ Receptor lento.
- Otros factores:
  - ☐ Realimentación.
  - ☐ Control de flujo versus control de congestión.



## 4.1 Requerimientos de QoS Voz

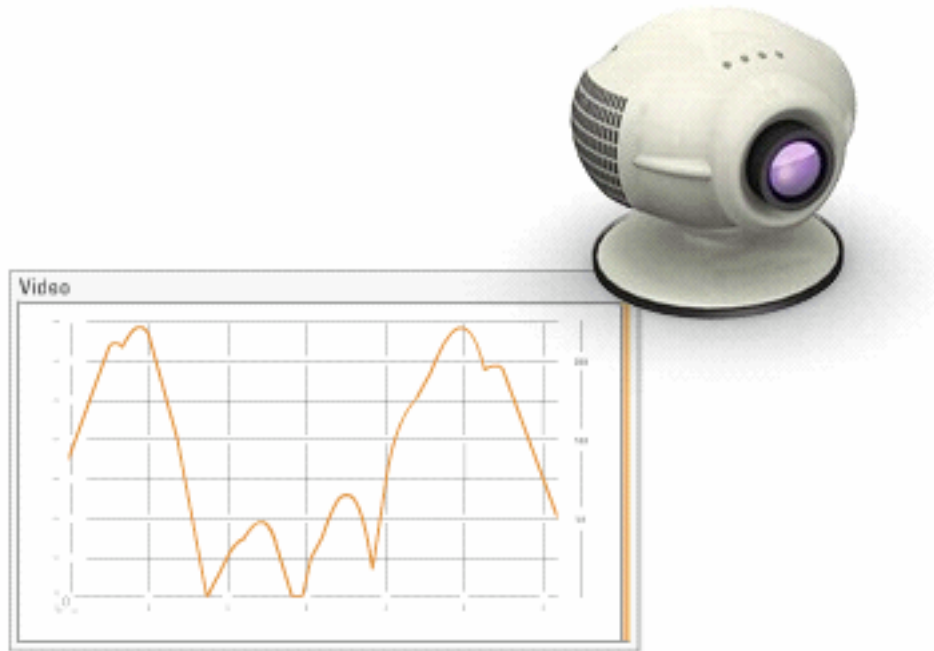


- Latencia < 150ms\*
- Jitter < 30ms\*
- Pérdidas < 1%\*
- 17-106 kbps garantizados por llamada.
- 150 bps garantizados de control por llamada.

\* requisitos en un solo sentido.

Suave, benigno, sensible a las pérdidas y al retraso, prioridad UDP.

## 4.1 Requerimientos de QoS Video



Ráfagas, benigno, sensible a las pérdidas y al retraso, prioridad UDP.

- Latencia < 150ms
  - Jitter < 30ms\*
  - Pérdidas < 1%\*
  - 17-106 kbps garantizados por llamada.
  - El ancho de banda mínimo garantizado es:
    - ❑ Stream de video + 20%
    - ❑ Por ejemplo, 384 kbps stream requiere 460kbps.
- \* requisitos en un solo sentido.



## 4.1 Requerimientos de QoS

Aplicación	Confiable	Retraso	Jitter	Ancho de banda
Correo	alto	bajo	bajo	bajo
Transferencia de ficheros	alto	bajo	bajo	medio
Acceso Web	alto	medio	bajo	medio
Login remoto	alto	medio	medio	bajo
Audio bajo demanda	bajo	bajo	alto	medio
Video bajo demanda	bajo	bajo	alto	alto
Telefonía	bajo	alto	alto	bajo
Videoconferencia	bajo	alto	alto	alto

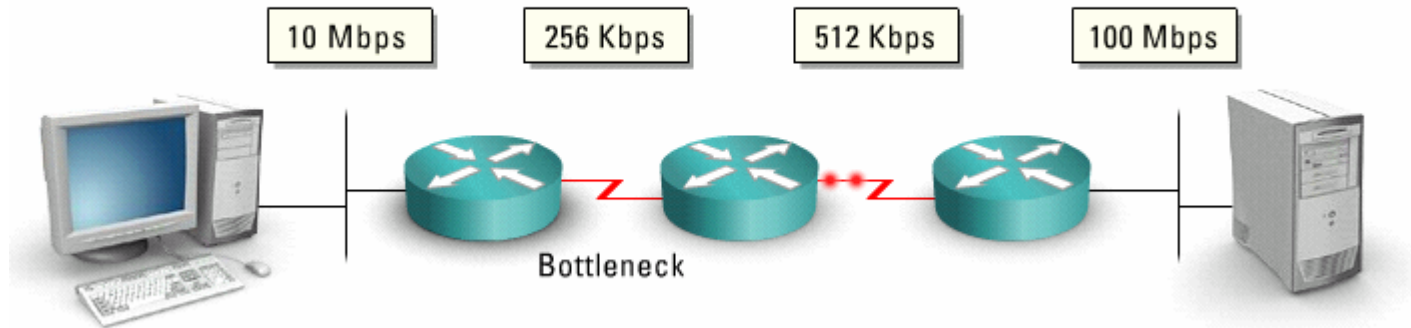


## 4.1 Definición de QoS

- **QoS, *Quality of Service***. Es la capacidad de proporcionar diferente prioridad a diferentes aplicaciones, usuarios, o flujos de tráfico, o garantizar un cierto nivel de rendimiento al flujo. Es necesario, como se ve en la tabla anterior, tratar flujos de tráfico de manera diferenciada, en función de su propósito.
- En **ausencia** de congestión de red, los mecanismos de QoS no son requeridos.
- De no hacerse, al compartir la misma infraestructura y recursos limitados, las aplicaciones que usan la red no cumplirían su propósito.
- P.e. el flujo de tráfico de una transferencia de ficheros, paquetes grandes provocarían retraso variable en la salida de los interfaces, y por tanto un *jitter* inaceptable.
- Las redes de transporte de operador tienen sus propios sistemas de QoS, como por ejemplo ATM (VBRrt, CBR, VBRnrt, ABR, UBR) o MPLS (EXP bit en IP).



## 4.1 Requerimientos de QoS



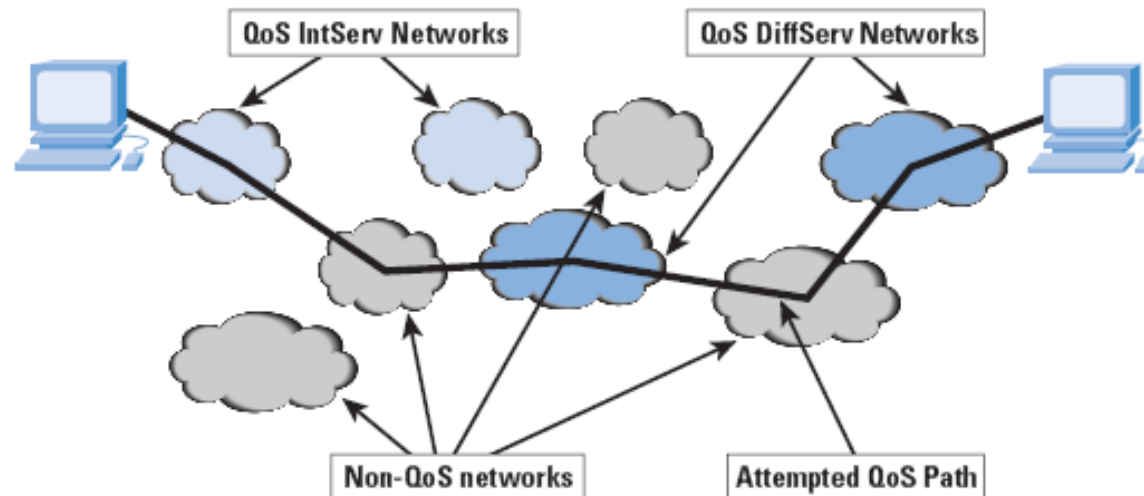
$$\text{Bandwidth}_{\max} = \min (10\text{Mbps}, 512\text{kbps}, 256\text{kbps}, 10 \text{ Mbps}) = 256 \text{ kbps}$$

- El ancho de banda máximo disponible es igual al menor de los enlaces.
- Muchos flujos compiten por el mismo ancho de banda disponible, resultando mucho menos disponible para cada aplicación.

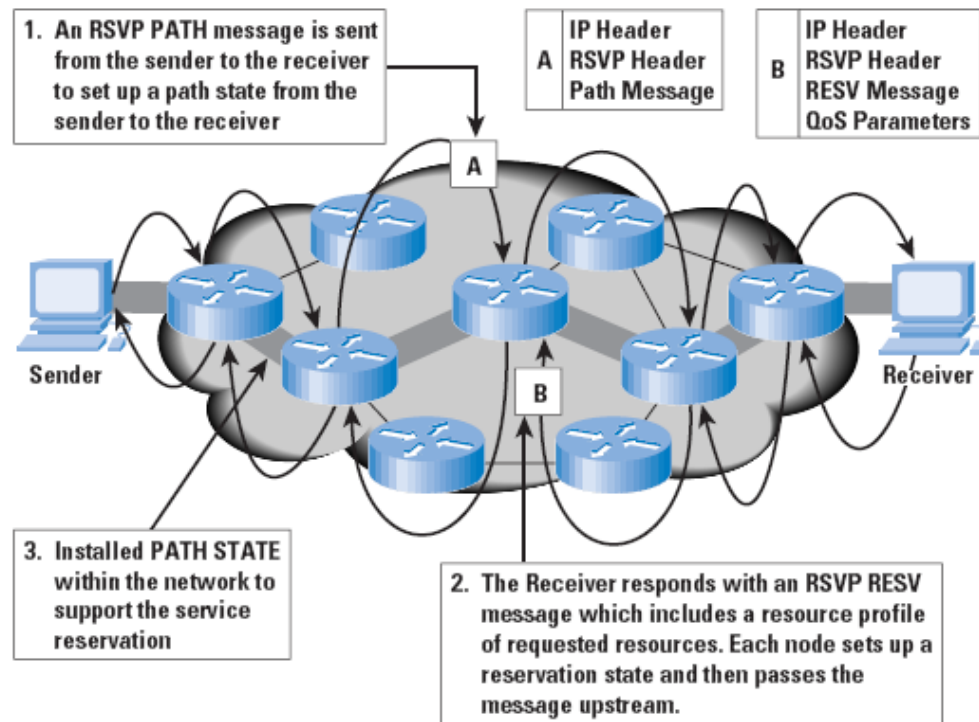


## 4.1 Arquitecturas de QoS del IETF

- **Diffserv.** Arquitectura de Servicios Diferenciados. Usada de forma más general. Más simple de implementar ya que se hace nodo a nodo. Usada en redes empresariales.
- **Intserv.** Arquitectura de Servicios Integrados. Más compleja. Es extremo a extremo. Implica un protocolo de señalización entendido por los nodos. Usada en MPLS TE.

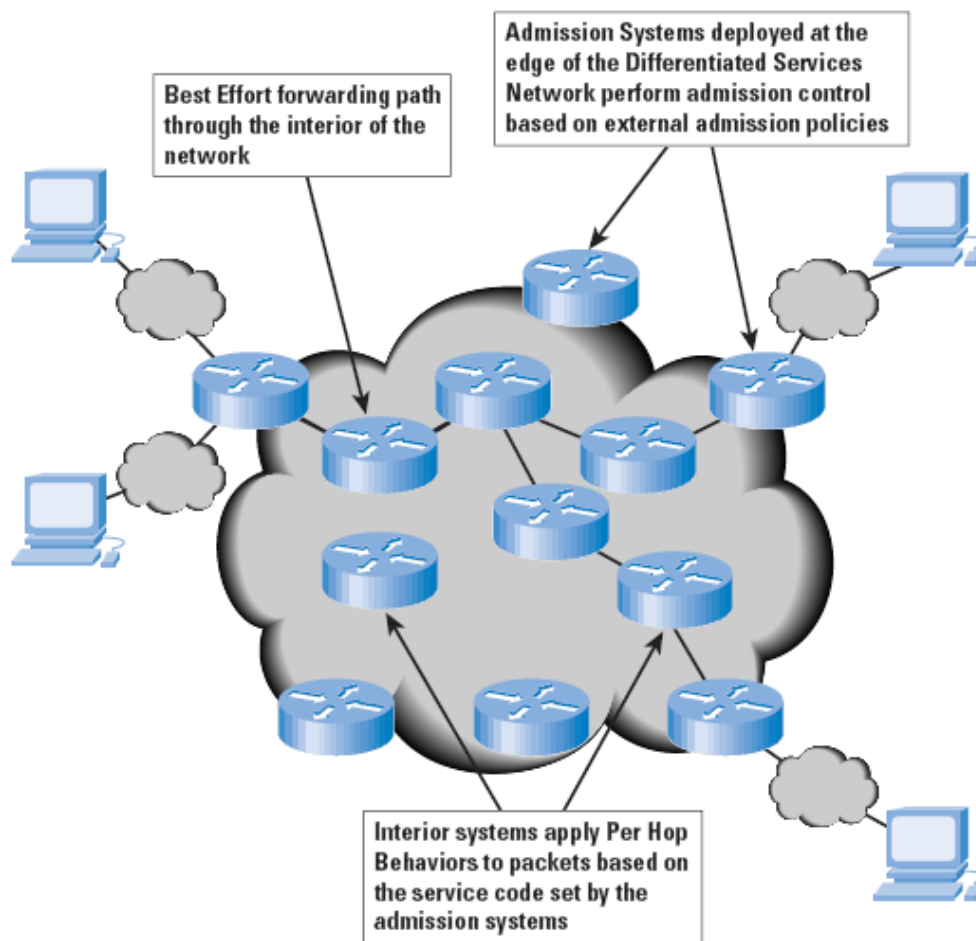


## 4.1 Arquitectura de QoS de Servicios Integrados IntServ



- El núcleo de IntServ es el protocolo de señalización, RSVP (Resource ReSerVation Protocol), para reserva de recursos para servicios.
- La implementación de las garantías se hace utilizando técnicas de priorización y desencolamiento.
- Descrita en la RFC 1633.

## 4.1 Arquitectura de QoS de Servicios Diferenciados Diffserv



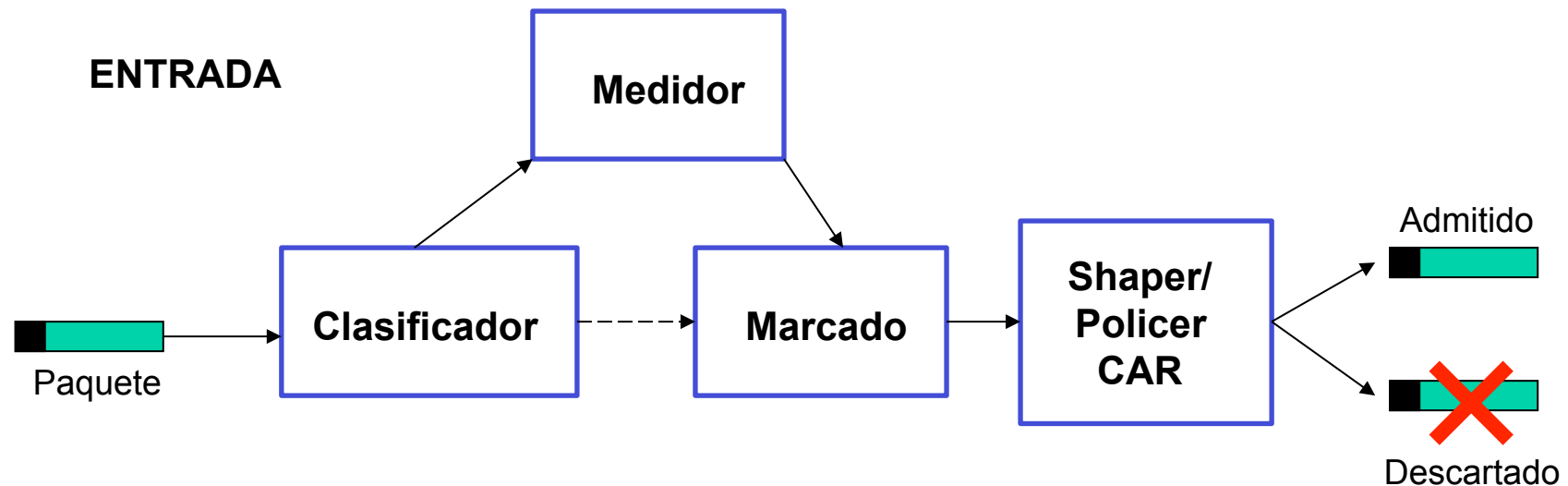
- El punto clave es la clasificación de los tipos de tráfico usando el campo DSCP. Los paquetes se clasifican (criterios) y se marcan, en el borde de la red. Después se tratan con técnicas de *policing*, conformado y priorización o desencolamiento.
- Este tratamiento se da salto a salto (PHB *Per Hop Behavior*). Los recursos se organizan para atender a los diferentes flujos según su prioridad.
- Descrita en RFC 2475.

## 4.1 Componentes de la arquitectura de QoS



- Clasificación
- Marcado (*Coloring*)
- Control de admisión (entrada)
- Conformación y Policing de tráfico (salida)
- Gestión de la congestión
- Evitación de la congestión (*Congestion Avoidance*)
- Eficiencia del enlace
- Señalización
  
- Sobre-provisión

## 4.1 Esquema de clasificación y admisión



- El clasificador inspecciona los paquetes, y los marca o colorea.
- La clasificación se realiza en función direcciones origen y o destino, protocolos, u otros campos del paquete.
- Cada paquete pertenece a una clase de red.



## 4.1 Mercado ToS

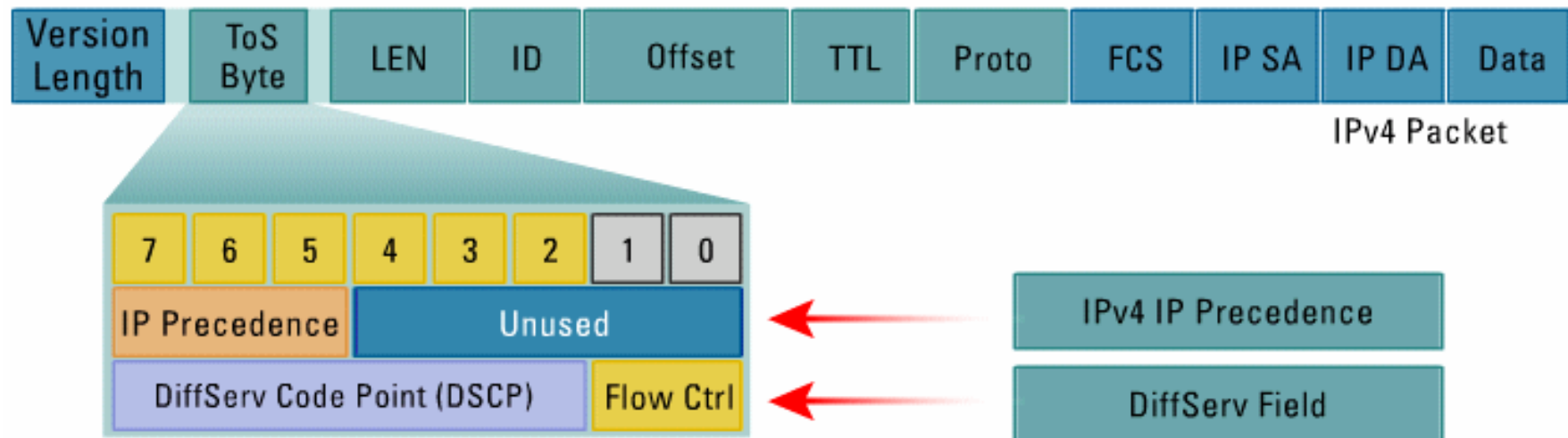
- En el capítulo 2.1 vimos lo siguiente:

- ☐ [2.2 TOS](#)

- ☐ [2.2 TOS Precedencia](#)

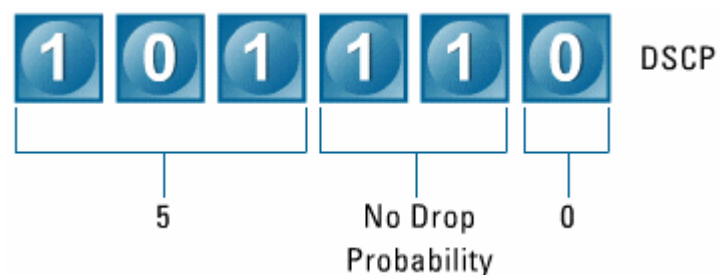
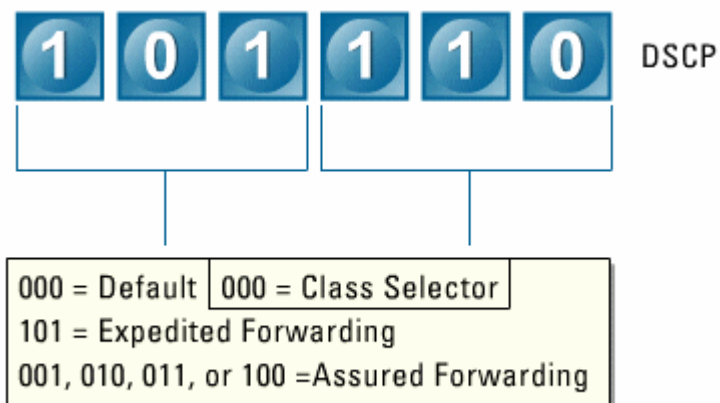
## 4.1 Marcado DSCP

- **DSCP** *Differentiated Services Code Point* o Campo *DiffServ* . Definido en la RFC 2474.
- Los 6 primeros bits se usan para seleccionar un método de reenvío y desencolamiento (PHB).





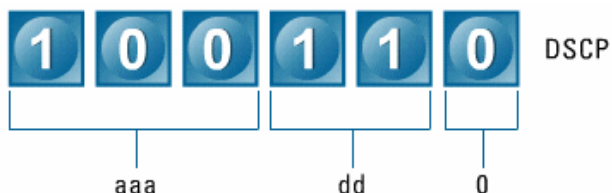
## 4.1 Marcado DSCP EF



- DSCP selecciona el PHB:
  - ☐ Comportamiento por defecto (FIFO y *Tail Drop*).
  - ☐ *Expedited Forwarding* EF PHB.
  - ☐ *Assured Forwarding* AF PHB.
  - ☐ *Class Selector* (IP Precedence) PHB
- *Expedited Forwarding* EF PHB:
  - ☐ Asegura un ratio mínimo de partida.
  - ☐ Garantía de ancho de banda, con envío priorizado.
  - ☐ Policía de ancho de banda, el exceso se tira.
- DSCP 101110. Precedencia 5.



## 4.1 Marcado DSCP AF



Class	Value		
AF1	001	dd	0
AF2	010	dd	0
AF3	011	dd	0
AF4	100	dd	0

Drop Probability (dd)	Value	AF Value
Low	01	AF11
Medium	10	AF12
High	11	AF13

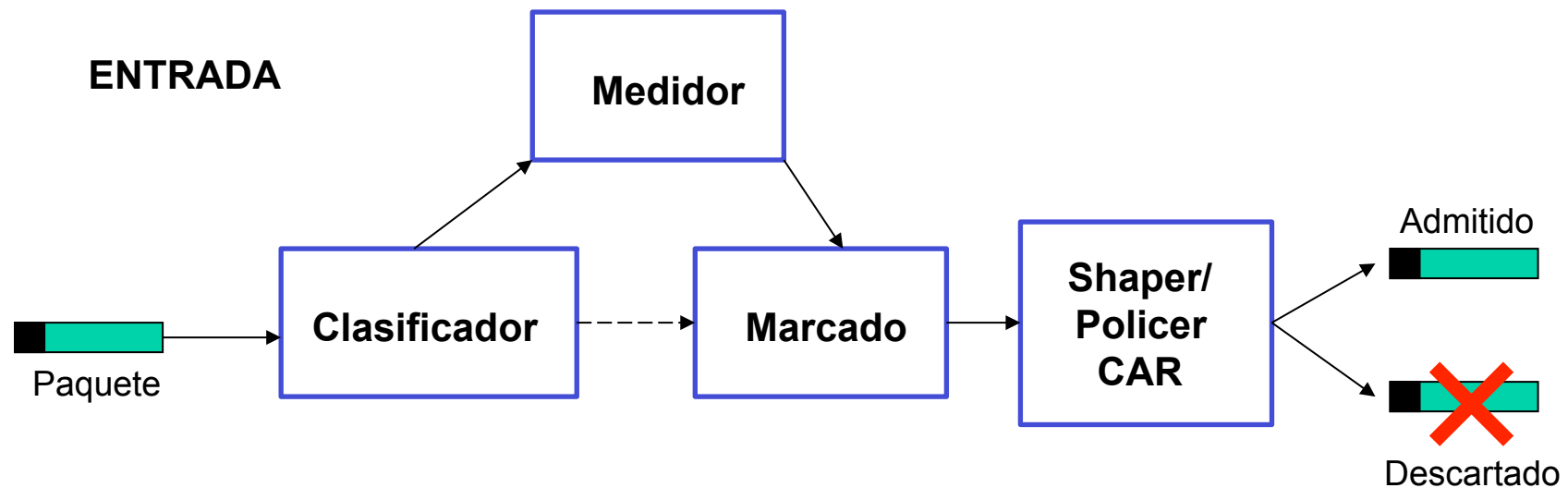
- *Assured Forwarding* AF PHB
  - ❑ Ancho de banda garantizado
  - ❑ Permite el acceso a ancho de banda extra si está disponible.
- Cuatro clases: AF1, AF2, AF3 y AF4.
- Rango de DSCP “aaadd0”
  - ❑ aaa valor binario de la clase.
  - ❑ dd es la probabilidad de descarte.
- Cada clase se reenvía de forma independiente según su ancho de banda garantizado.
- RED (*congestion avoidance*) es utilizado para clase.

## 4.1 Clasificación y Marcado L3, L2 y MPLS EXP

Application	L3 Classification			L2 CoS	L2 MPLS EXP
	IPP	PHB	DSCP		
Reserved	7	-	56-63	7	7
Reserved	6	-	48-55	6	6
Voice Bearer	5	EF	46	5	5
Videoconferencing	4	AF41	34	4	4
Mission-Critical Data	3	AF31	26	3	3
Transactional Data	2	AF2X	18,20,22	2	2
Bulk Data	1	AF1X	10,12,14	1	1
Best-Effort Data	0	BE	0	0	0
Less-than-Best-Effort Data	0	-	2,4,6	0	0

## 4.1 Control de admisión

- El **control de admisión** utiliza *Traffic Policing* para regular la entrada de paquetes a través de un interfaz.
- Trata de eliminar el tráfico excedente de acuerdo al contrato, lo más cerca de la fuente.
- Usado por los operadores en ingreso o perímetro de redes MPLS.







CEU

## 4 La capa de red II

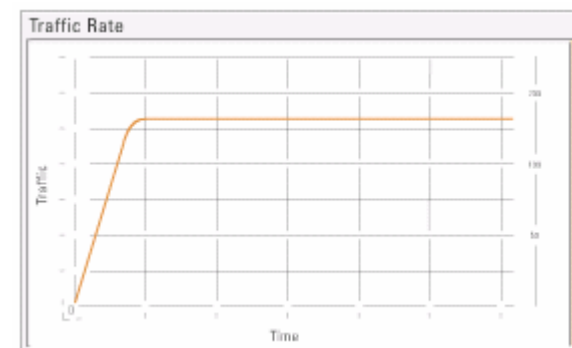
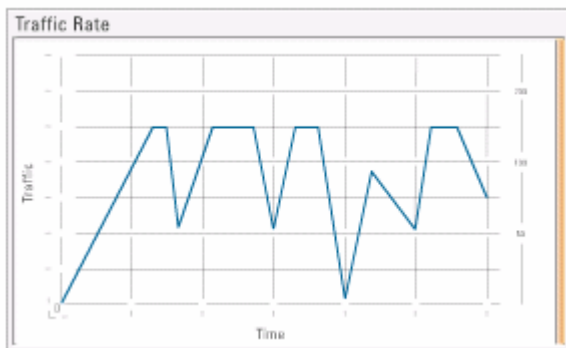
### 4.1 Calidad de Servicio (2/2)





## 4.1 Conformación y *Policing* de tráfico

- ***Traffic Shaping y Traffic Policing*** son usados para producir patrones de tráfico predecibles.
- Ambas técnicas eliminan ráfagas. Por esto es más adecuado para redes de datagramas.



### ***Traffic Policing o Rate Limiting***

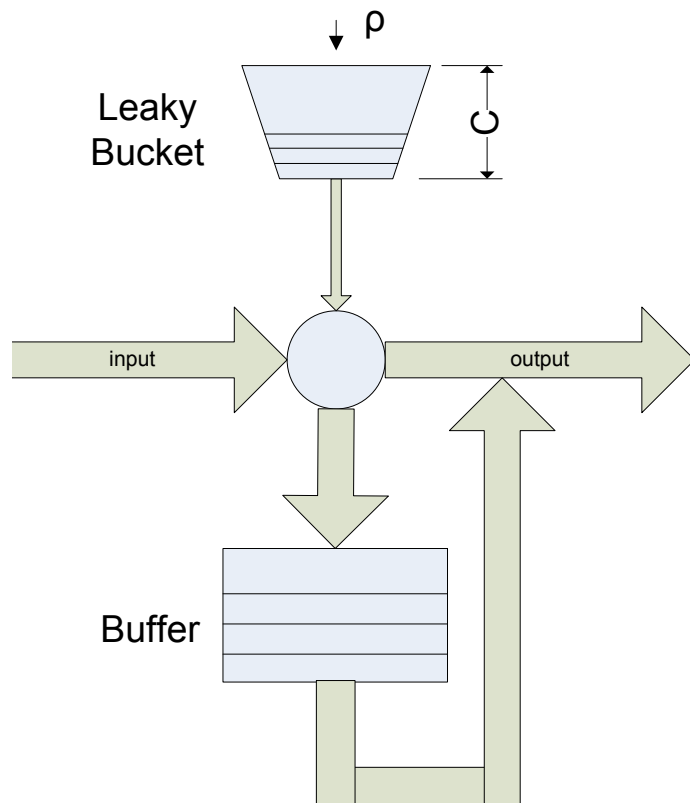
- Entrada y Salida.
- Paquetes excedentes se descartan.
- Descartes causan retransmisión TCP.
- Usa menos buffer.

### ***Traffic Shaping***

- Entrada y Salida (depende equipo).
- Paquetes excedentes se encolan hasta el llenado del buffer.
- Minimiza retransmisión TCP.
- Integración con Frame-Relay.



## 4.1 Algoritmo de cubeta con goteo

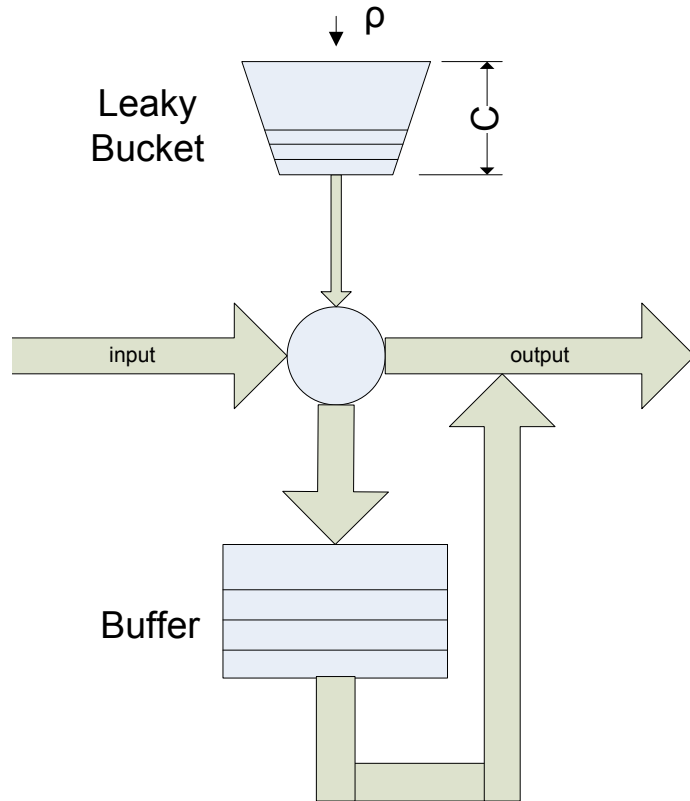


- El algoritmo de **Leaky bucket** o cubeta con goteo sirve para implementar *traffic shaping* o *rate limiting* en salida.
- Los parámetros del algoritmo son:
  - ❑  $\rho$ , la velocidad de salida de cubo.
  - ❑  $C$ , la capacidad del cubo. Expresa la capacidad de la ráfaga sin descarte.
- GCRA, *General Cell Rate Algorithm*, usado por ATM para hacer conformado de tráfico [4].
- Aunque las velocidades se expresen en b/s, el algoritmo se encarga de hacer la alineación a paquetes



## 4.1 Algoritmo de cubeta con goteo

- Proceso:
  - ☐ Inicialmente el cubo está vacío.
  - ☐ Por cada paquete que llega, se incrementa el contador de llenado del cubo.
  - ☐ De forma programada,  $\rho$ , se decrementa el contador de llenado del cubo.
  - ☐ Si el cubo no está lleno, el paquete se transmite.
  - ☐ Si el cubo está lleno, el paquete se descarta.
- La relación que nos da el tamaño del cubo es la siguiente. Donde:
  - ☐  $r$ , es el tamaño de la ráfaga a soportar b/s.
  - ☐  $t$ , es el tiempo de la ráfaga.



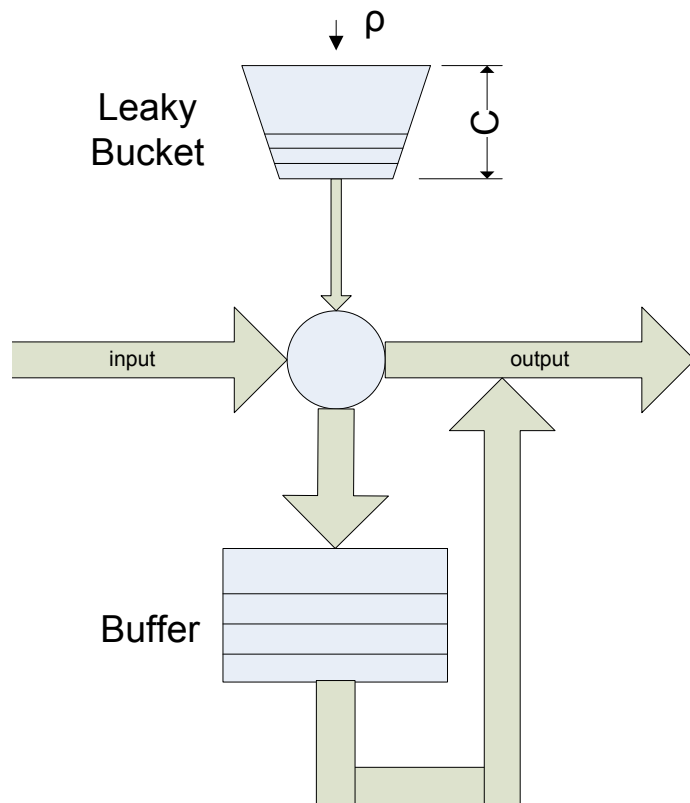
$$C \geq rt - \rho t \therefore \rho = \frac{B_c}{t_c}$$

$$R = rt = C + \rho t$$

R: tamaño de la ráfaga, en bits o Bytes.



## 4.1 Algoritmo de cubeta con goteo



- Este algoritmo implementa la función de:
  - ☐ Conformación de Tráfico. Comienza a descartar paquetes cuando se agota la cubeta.
  - ☐ Rate limiting. Se consigue con  $C=1$ . Comienza a descartar paquetes inmediatamente.
- No sirve para implementar en entrada shaping/policing, pues los paquetes son descartados.



## 4.1 Algoritmo de cubeta con goteo Ejemplo

- Cubeta de  $\rho = 1$  MB/s, capacidad  $C = 1$  MB.
- ¿Durante cuánto tiempo podrá soportar una ráfaga de 2MB?

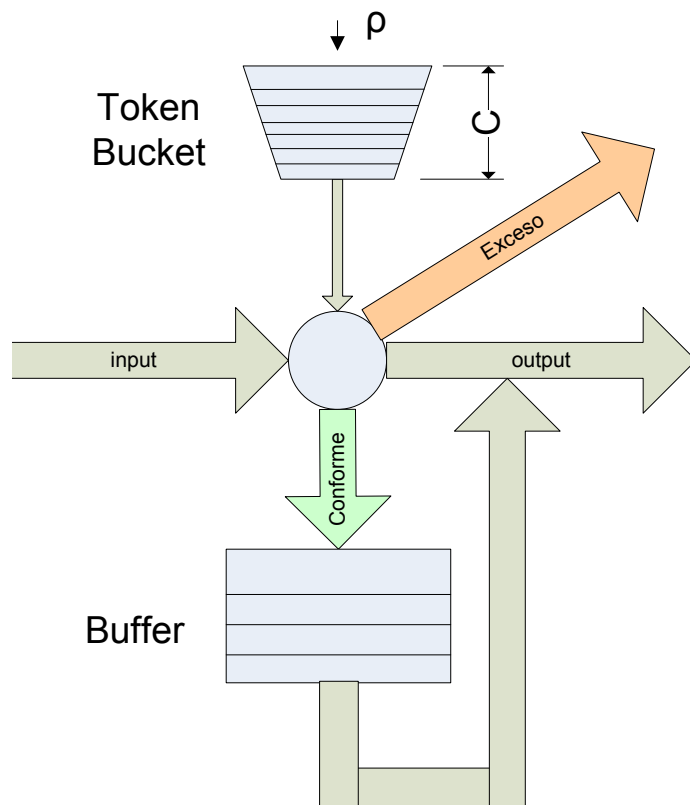
$$1MB = 2MB - 8Mb/s \cdot t \therefore t = \frac{8Mb}{8Mb/s} = 1s$$

- Podría manejar ráfagas de 2 MB durante 1000 ms.





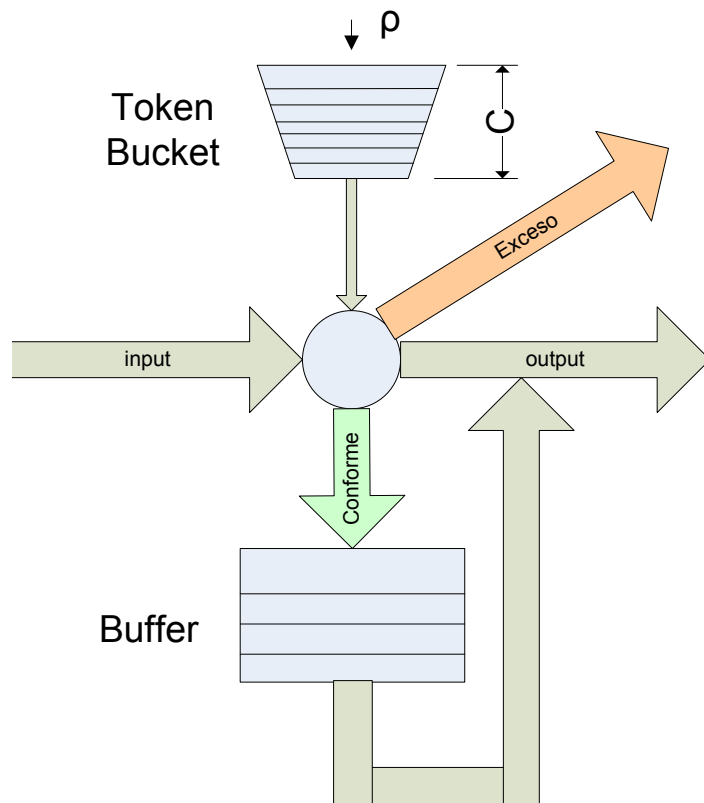
## 4.1 Algoritmo de cubeta con ficha



- El algoritmo de **Token bucket** o cubeta con **ficha** sirve para implementar *traffic shaping* o *rate limiting* en entrada y salida.
- Los parámetros del algoritmo son:
  - ❑  $\rho$ , la velocidad de entrada de fichas.
  - ❑  $C$ , la capacidad del cubo. Expresa la capacidad de la ráfaga inicial.
- Aunque las velocidades se expresen en b/s, el algoritmo se encarga de hacer la alineación a paquetes



## 4.1 Algoritmo de cubeta con ficha



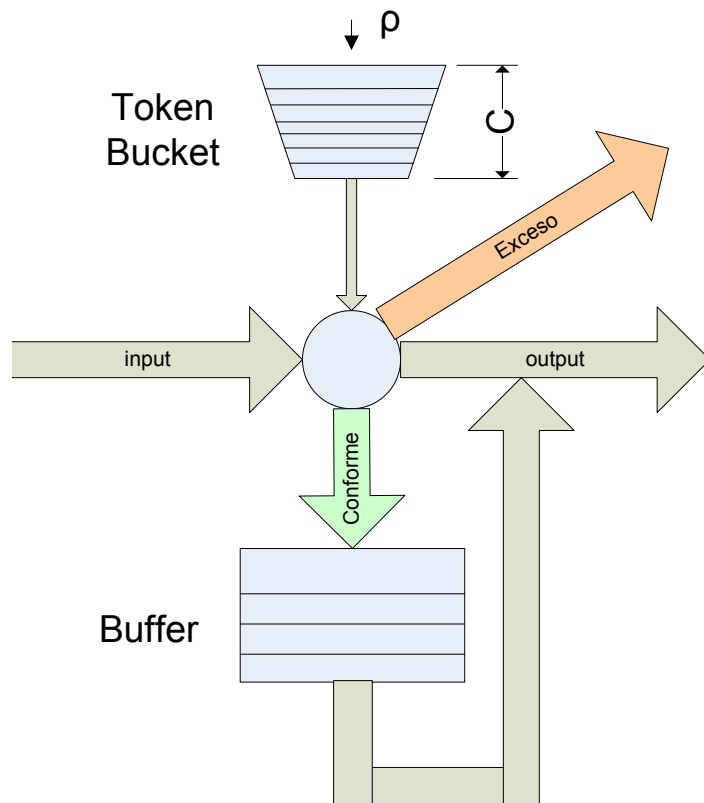
- Proceso:
  - ☐ Inicialmente el cubo de fichas está lleno.
  - ☐ De forma programada,  $\rho$ , el cubo se llena de fichas.
  - ☐ Si el cubo está lleno, la ficha se descarta.
  - ☐ Por cada paquete que sale, se decrementa un contador de llenado del cubo.
  - ☐ Si el cubo está vacío, el paquete entra en una acción especial (exceso).
- Dado que inicialmente el cubo está lleno, permite enviar paquetes a la velocidad del enlace.
- Posteriormente se enviará a la velocidad de entrada de fichas.



## 4.1 Algoritmo de cubeta con ficha

- La ráfaga inicial se puede expresar con la fórmula siguiente. Donde:

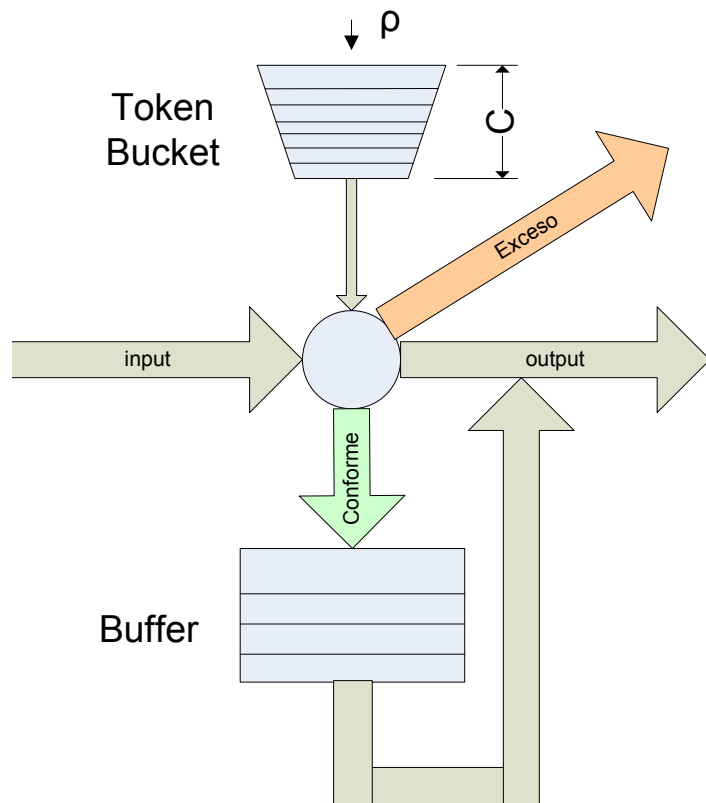
- ☐ M, velocidad de salida interfaz.
- ☐ t, tiempo de la ráfaga inicial.



$$Mt = C + \rho t \therefore \rho = \frac{B_c}{t_c}$$



## 4.1 Algoritmo de cubeta con ficha

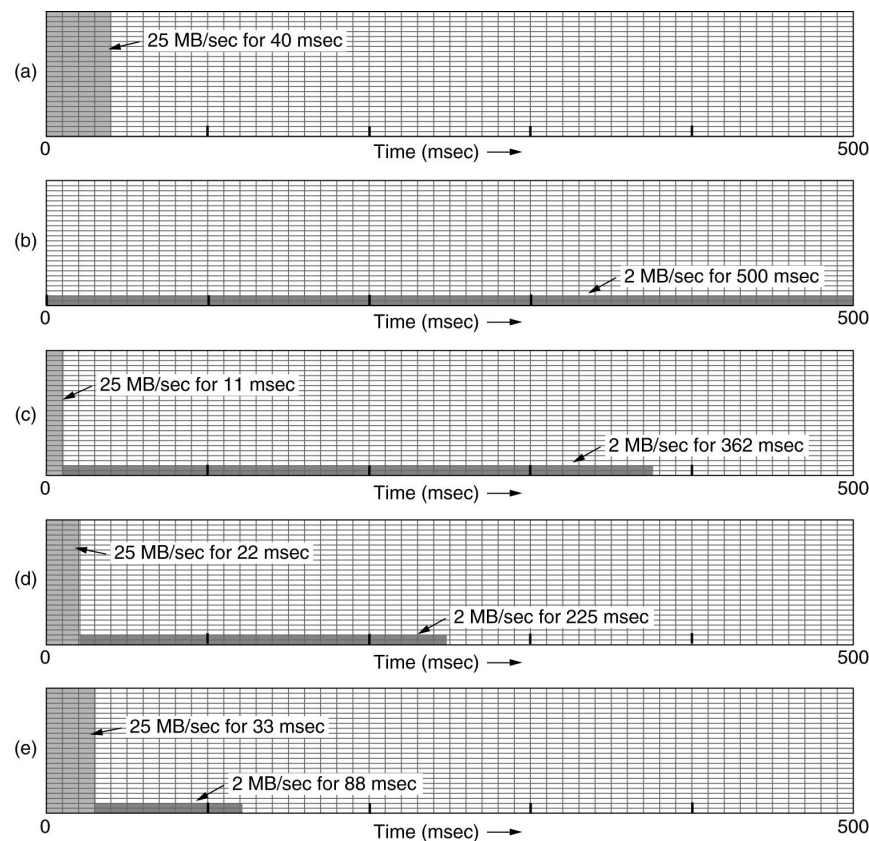


- Este algoritmo implementa la función de:
  - ☐ Conformación de Tráfico.  $C=1$ . Acción con el exceso, encolar FIFO. Conformado a  $\rho$ .
  - ☐ Rate limiting.  $C=1$ . Acción con el exceso, descartar. Límite de  $\rho$ .
  - ☐ Ídem con ráfaga.  $C>0$ .
- Las acciones con el exceso pueden ser:
  - ☐ Descartar (Policy).
  - ☐ Remarcado y envío a otro Token Bucket. P.e. implementación CIR y EIR en Frame-Relay.



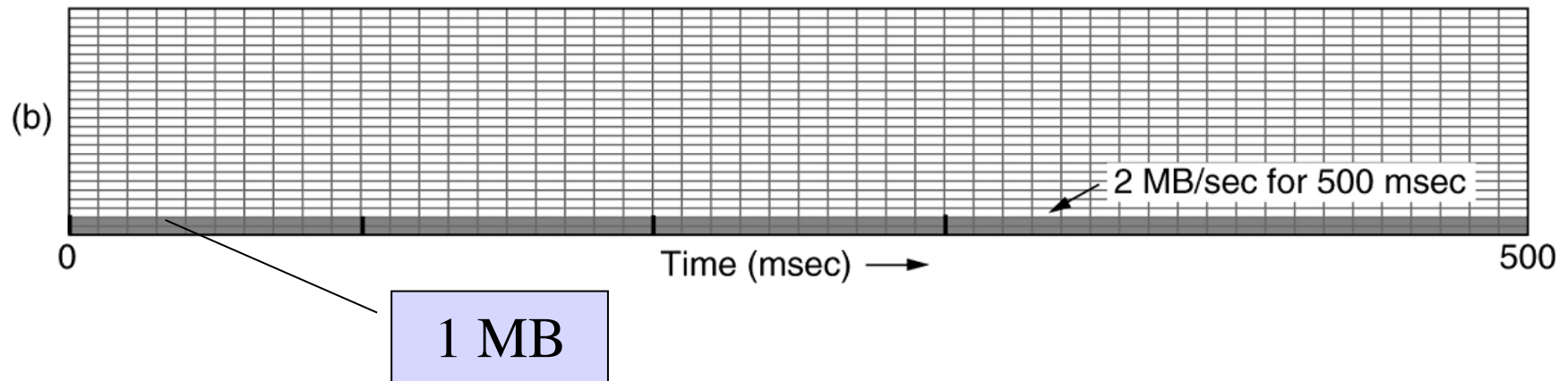
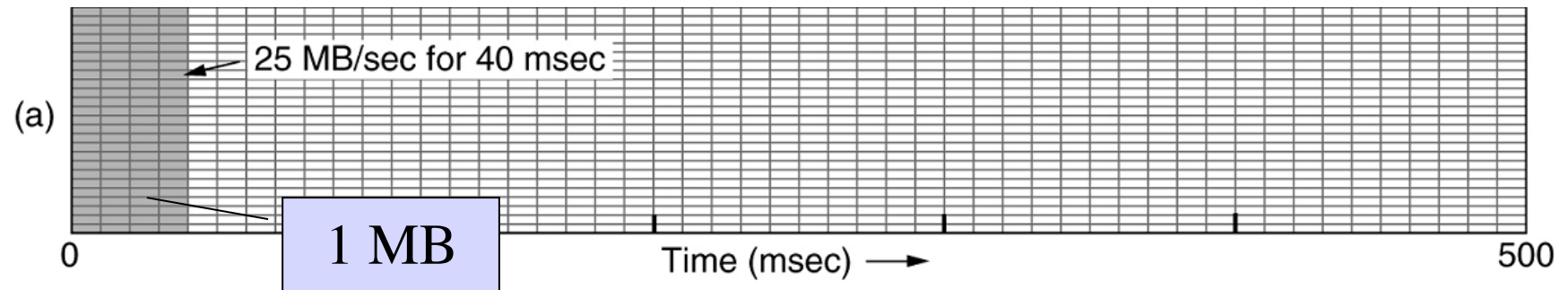
## 4.1 Algoritmo de cubeta con ficha Ejemplo

- a) Entrada a cubeta con goteo.
- b) Salida de la cubeta con goteo. Salida de la cubeta con fichas con capacidades de:
- c) 250 KB,
- d) 500 KB,
- e) 750 KB,





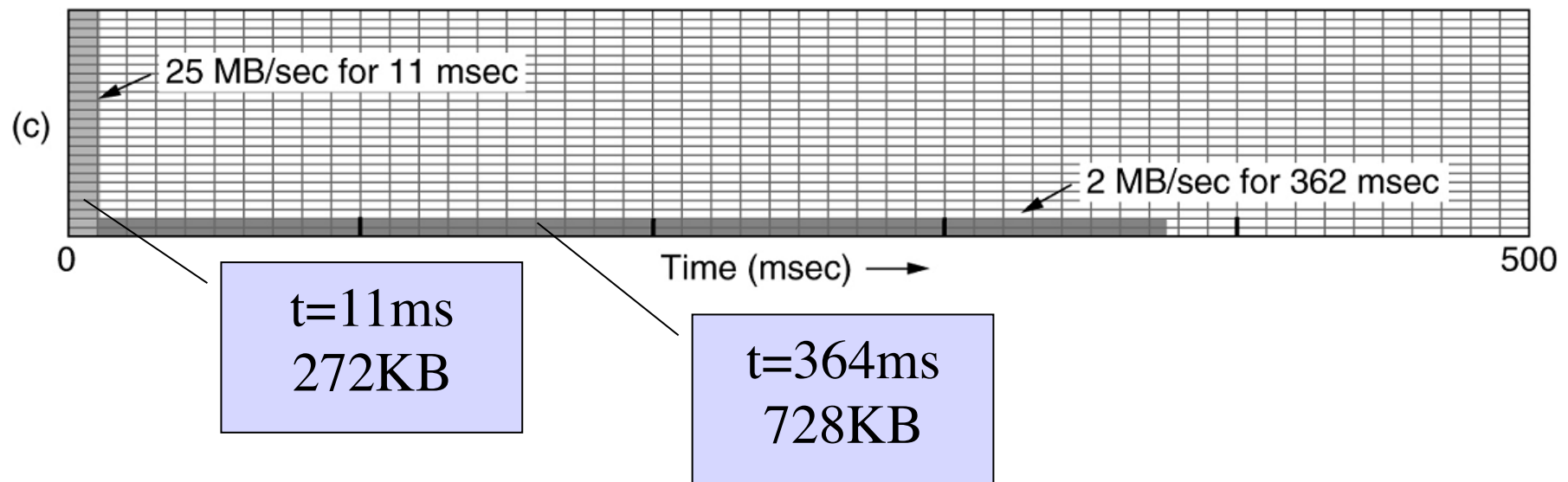
## 4.1 Algoritmo de cubeta con goteo Ejemplo







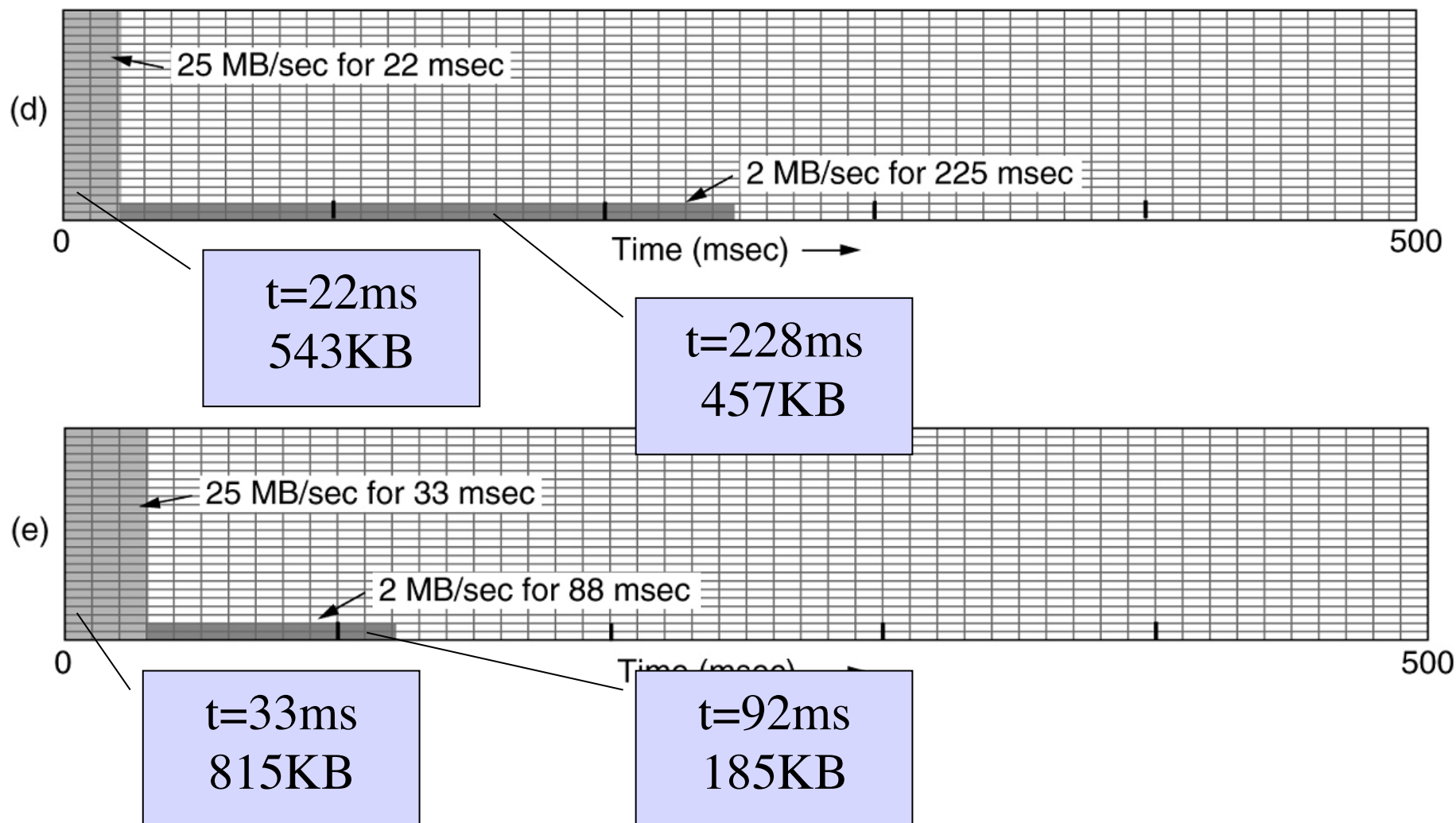
## 4.1 Algoritmo de cubeta con ficha Ejemplo



$$t = \frac{C}{M - \rho}$$



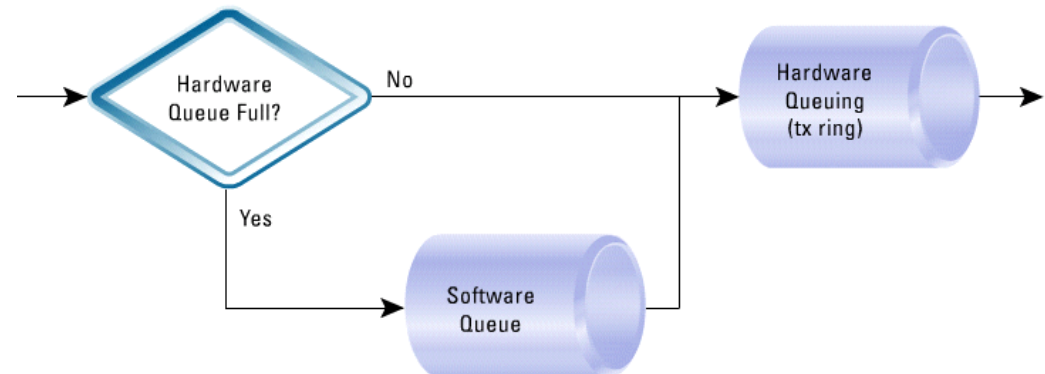
## 4.1 Algoritmo de cubeta con ficha Ejemplo



## 4.1 Gestión de la Congestión

Técnicas de gestión de la congestión:

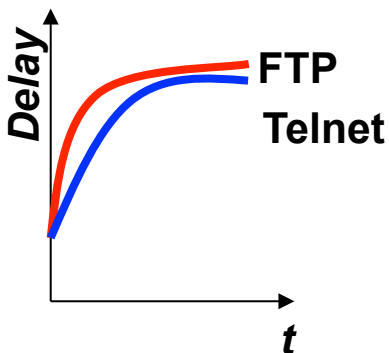
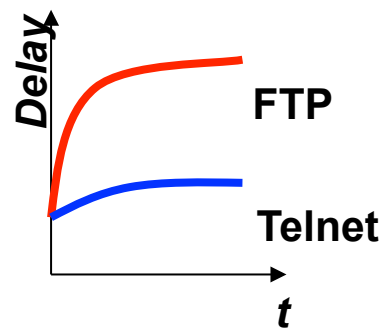
- ☐ FIFO.
- ☐ FQ.
- ☐ WFQ.
- ☐ CBWFQ.
- ☐ PQ.
- ☐ PQ+CBWFQ.
- ☐ CQ (DWRR).



- La gestión de la congestión no elimina la congestión permanente. Debe ser usada puntualmente.
- La cola hardware debe ser muy pequeña. Si hay congestión debe haber otros mecanismos.



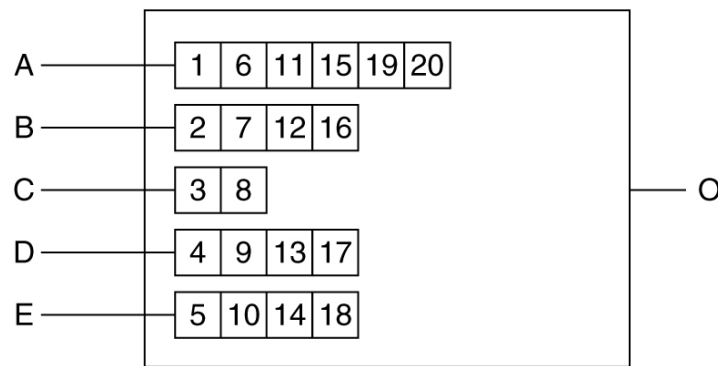
## 4.1 Fair Queueing



- **FIFO**, Primero en llegar, primero en salir. Tiene el severo inconveniente que todos los flujos (parejas de IP origen, IP destino), reciben el mismo tratamiento. Los flujos de tiempo real, paquetes habitualmente pequeños, reciben un mal condicionamiento (variación del retraso).
- **FQ**, encolamiento justo. Reparte el ancho de banda de forma equitativa. Establece tantas colas como flujos. Establece un *max-min fairness*. Es decir, transmite primero la cola que tiene menor ratio de datos, a continuación la segunda con menor ratio, y así sucesivamente.
- FQ tiene menor *throughput*, pero evita los efectos de inanición provocados por los flujos con mayor consumo.

## 4.1 Fair Queueing

- FQ selecciona el orden de transmisión de los paquetes modelando el orden de terminación de cada paquete, como si se pudieran enviar bit a bit, en *Round Robin*.
- El paquete con tiempo de terminación menor va primero, y así sucesivamente. Realmente se usa un algoritmo modificado para reducir el gasto computacional.



(a)

Packet	Finishing time
C	8
B	16
D	17
E	18
A	20

(b)

(a) Un router con 5 paquetes para sacar por la línea O.

(b) Tiempos de servicio para los 5 paquetes.

## 4.1 WFQ *Weighted Fair Queueing*

- **WFQ**, encolamiento justo ponderado, es una generalización de FQ.
- FQ asigna la misma prioridad a todos los flujos.
- WFQ asigna prioridades diferentes a flujos caracterizados de forma diferente. Esta caracterización puede ser la DSCP.
- Los repartos de ancho de banda son para FQ y WFQ:

$$\frac{R}{N}$$

$$\frac{Rw_j}{\sum_i w_i}$$

□ R: Es el ancho de banda total.

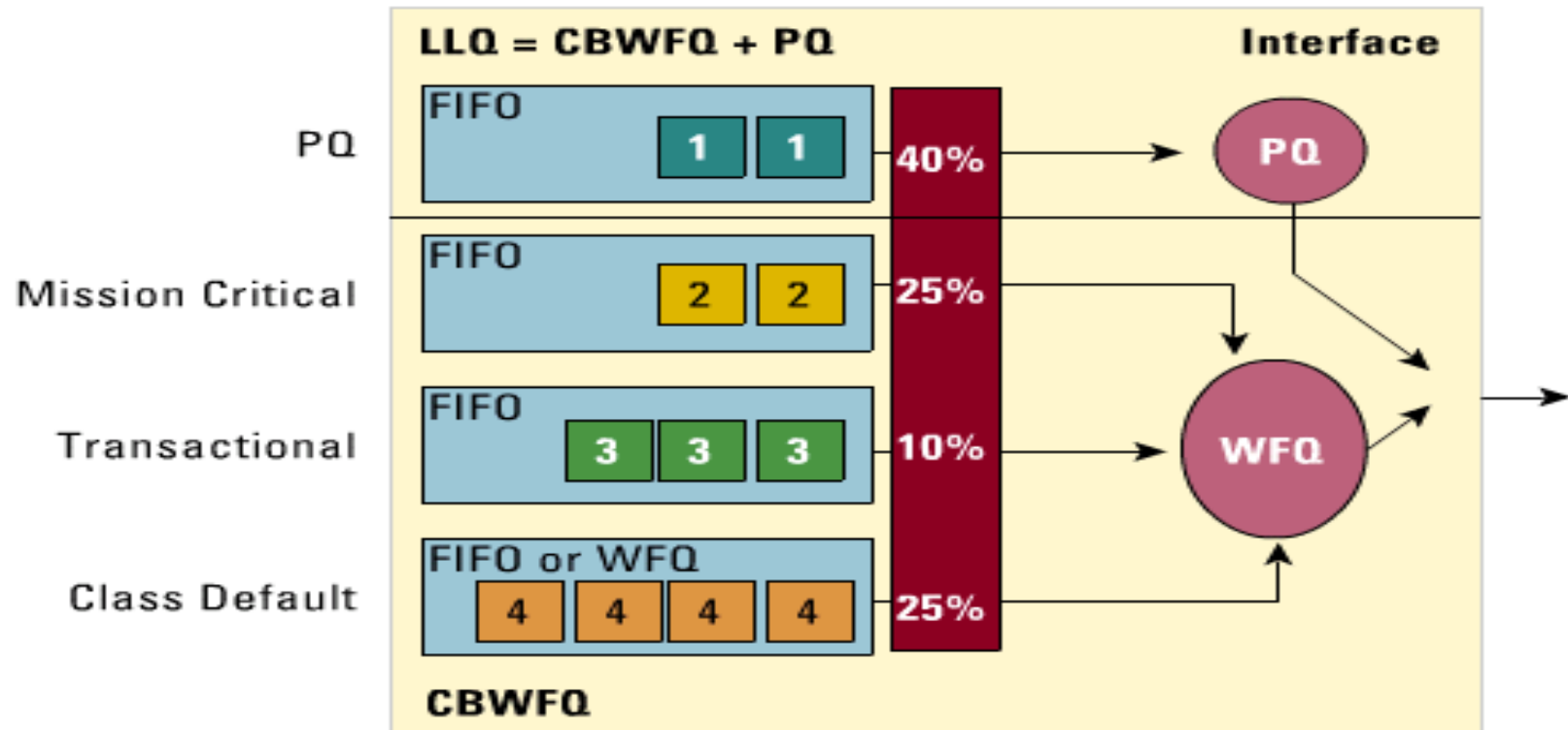
□ N: número de flujos.

□  $w_i$ : ponderación para un flujo  $i$ .

## 4.1 CBWFQ, PQ, LLQ

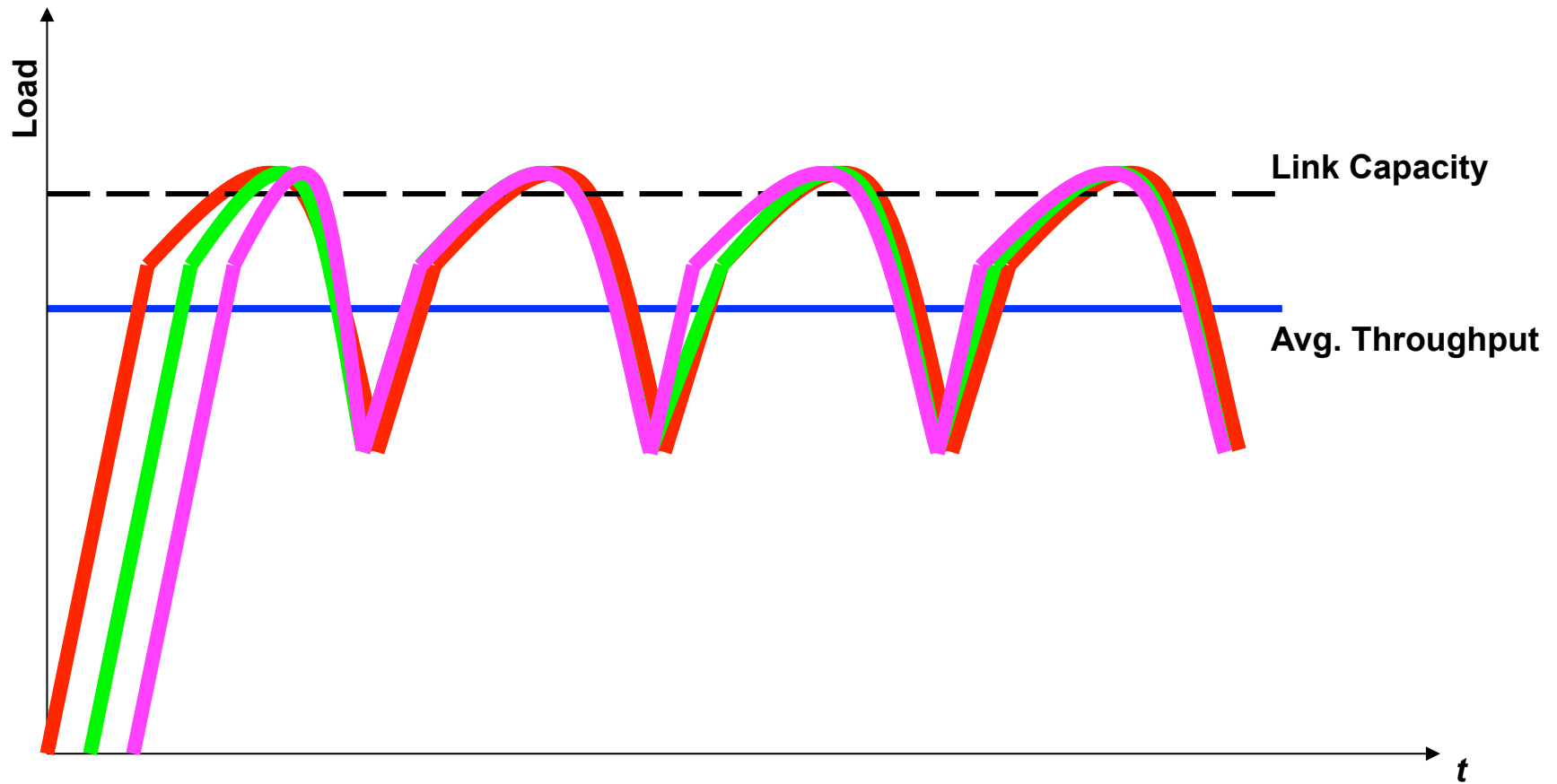
- **CBWFQ**, *Class Based WFQ*, o WFQ basado en clase, consiste en formar las diferentes colas de WFQ en función de la clasificación de paquetes en clases de red. En general una clases podrán desbordarse sobre otras si las últimas no están ocupadas y hay congestión.
- **PQ**, *Priority Queueing*, o encolamiento prioritario, consiste en dar prioridad absoluta a ciertos flujos de tráfico. Este tipo de programación debe venir limitada en ancho de banda para no provocar la inanición del resto.
- **LLQ**, *Low Latency Queueing* (Cisco), encolamiento de baja latencia, consiste en combinar CBWFQ con PQ.

## 4.1 CBWFQ, PQ, LLQ





## 4.1 Evitación de la congestión. Efecto de Sincronización global.



Efecto de sincronización global. Consecuencia del funcionamiento de TCP.

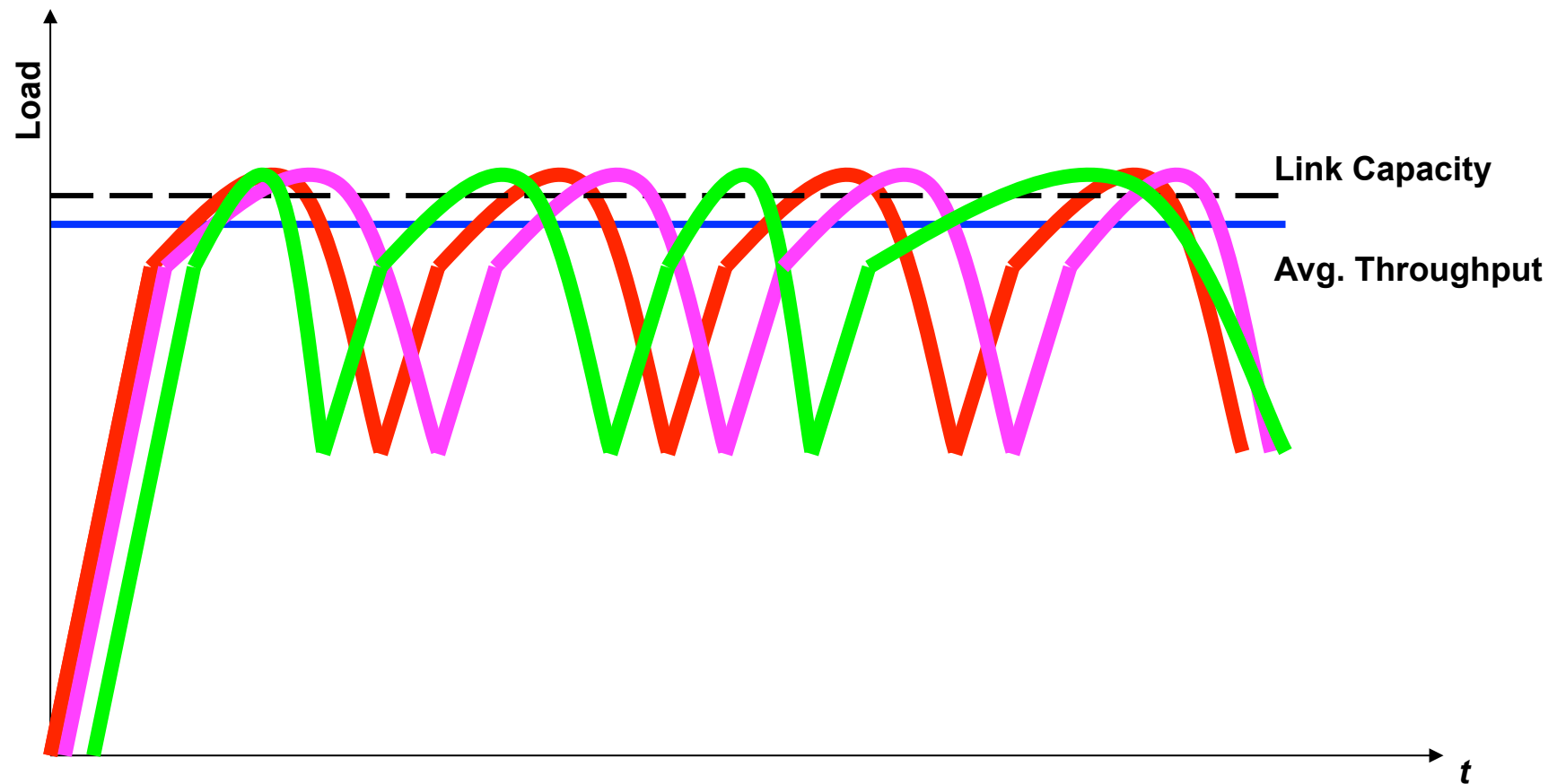
## 4.1 Evitación de la congestión. *Tail Drop* y control de flujo TCP.

- Como consecuencia del llenado de las colas (*Tail Drop*), se descartan paquetes de múltiples flujos simultáneamente.
- Existe una alta probabilidad de múltiples descartes en los flujos más consumidores en la misma sesión TCP.
- Los descartes se distribuyen uniformemente en los flujos interactivos.
- La consecuencia es: **Media de *Throughput* bajo.**
- *Tail Drop*, descartes al final, son los que se producen por el llenado de las colas.

## 4.1 Evitación de la congestión. RED.

- **RED.** *Random Early Detection*, es un método activo para evitar congestión desarrollado por Van Jacobson en 1993. Comienza a descartar aleatoriamente paquetes antes de que ocurra una congestión.
- Proceso:
  - ☐ Monitoriza la ocupación de la cola correspondiente al paquete entrante.
  - ☐ Si la cola supera un umbral, calcula la probabilidad de descarte.
  - ☐ Si la probabilidad es alta, descarta el paquete.
  - ☐ Si la cola está llena, se comporta como *tail drop*, y descarta.
  - ☐ En los demás casos, transmite el paquete.
- Los sistemas que más transmiten, tienen mayor probabilidad de que se descarten su paquetes.
- Consecuencias:
  - ☐ Mantiene baja la media de ocupación de las colas.
  - ☐ Incrementa la media del *Throughput*.
- **WRED**, *Weighted RED*, o RED ponderado, tiene en cuenta la prioridad (DSCP o la clase de red).

## 4.1 Evitación de la congestión. Efecto de Sincronización global.



Efecto de sincronización global eliminado. Consecuencia de RED.

## 4.1 Sobre-provisión

- La **sobre-provisión** es una estrategia aceptable.
- Consiste en utilizar más ancho de banda del necesario para evitar la congestión y por tanto, la necesidad de las técnicas anteriormente vistas.
- En la referencia [3] Odlyzko defiende que es una opción económicamente interesante.

## 4.1 Bibliografía

- [1] Tanenbaum, A. S., Computer Networks, 4ª Ed Pearson 2003, apartado 5.3 y 5.4.
- [2] Huston, Geoff, Quality of Service—Fact or Fiction?, Internet Protocol Journal 2000, volume 3 number 1.
- [3] Odlyzko, A., “The Economics of the Internet: Utility, Utilization, Pricing, and Quality of Service,” 1998. [www.research.att.com/~amo](http://www.research.att.com/~amo)
- [4] ATM Forum; Traffic Management Specification Version 4.1, 1999, pag. 24.