

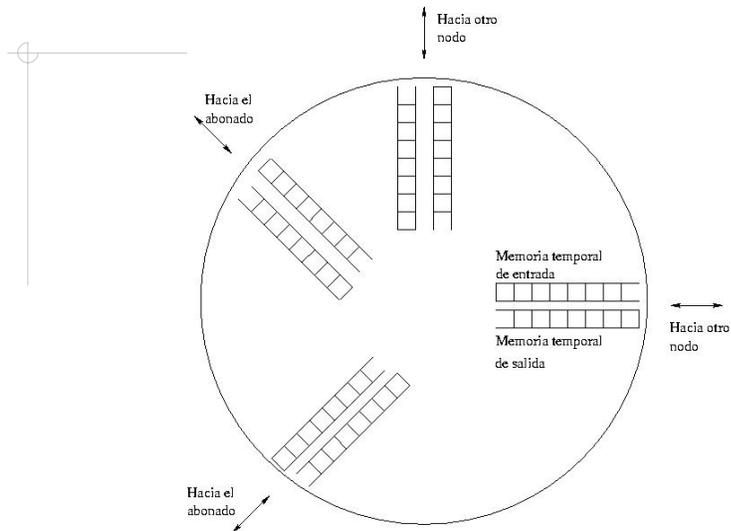
Control de Congestión

Redes y Servicios
de Comunicaciones I

Definición de Congestión

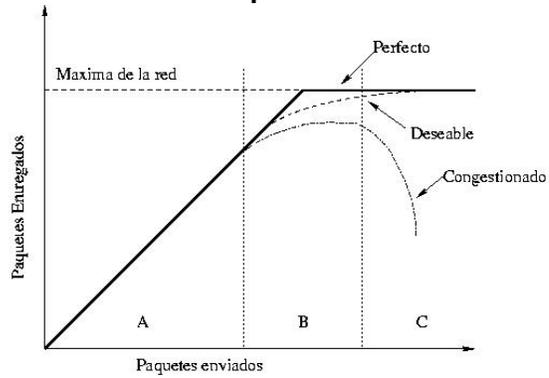
- ◆ Fenómeno producido cuando a la red (o parte de ella) se le ofrece más tráfico del que puede cursar.
 - Causa: Las memorias temporales de los nodos se desbordan.
 - ◆ Los paquetes se reciben demasiado deprisa para ser procesados (se llena memoria de entrada).
 - ◆ Demasiados paquetes en la memoria de salida esperando ser asentidos (se llena memoria de salida).

Colas en un nodo



Efectos de la congestión

Deterioro de las prestaciones de la red

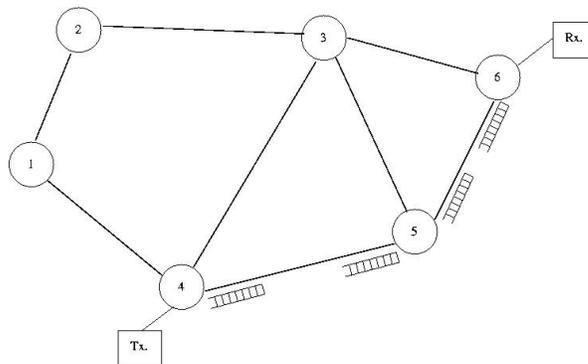


Estrategias

- ◆ Cuando se llega a la saturación (zona C):
 - Descartar cualquier paquete de entrada para el que no exista memoria.
 - Implementar algún tipo de control de flujo sobre sus vecinos de forma que el tráfico sea manejable.

Estrategias

- ◆ Problema:
 - Propagación de la congestión



Control de flujo vs Control Congestión

- ◆ Control de congestión: **intenta asegurar que la subred sea capaz de transportar el tráfico ofrecido.**
- ◆ Control de flujo: **tráfico punto a punto entre un transmisor y un receptor. Evita que un transmisor rápido sature a un receptor lento.**
 - **El control de flujo es una técnica más de control de congestión.**

Técnicas de Control de Congestión

- ◆ Bucle Abierto: **La idea es prevenir e intentar solucionar el problema antes de que se produzca. Para ello hay que diseñar la red de manera adecuada, actuando sobre diferentes parámetros a diferentes niveles.**
- ◆ Bucle Cerrado: **Son métodos reactivos, es decir, se actúa cuando aparece el problema, basándose en el presente estado de la red.**

Bucle Abierto

- ◆ Niveles en los que se actúa:
 - Transporte
 - ◆ Retransmitir si vencen los temporizadores
 - ◆ Descarte de tramas. Desorden en mensajes.
 - ◆ Control de flujo (ventanas).
 - Red
 - ◆ Descarte de paquetes.
 - ◆ C.V. vs Datagramas.
 - ◆ Algoritmos de encaminamiento: balanceo de carga entre líneas.
 - ◆ Tiempo de vida de los paquetes.
 - Enlace
 - ◆ Parecido a los anteriores, pero entre nodos.
 - ◆ Colas de los routers (teoría de colas).
 - ◆ Asentimientos: piggybacking

Bucle Cerrado

- ◆ Suelen hacerse en tres fases:
 - Monitorización: **para detectar cuándo y dónde sucede la congestión:**
 - ◆ Ocupación de enlaces y buffers.
 - ◆ Porcentaje de paquetes descartados.
 - ◆ Número de retransmisiones.
 - ◆ Retardos y jitter.
 - Reacción: **enviar información a los puntos en los que se pueda actuar contra la congestión.**
 - ◆ Enviar paquetes especiales a las fuentes.
 - ◆ Utilizar bits reservados en el campo de control del protocolo.
 - ◆ Enviar paquetes solicitando información explícita sobre congestión.
 - Ajustar la operación del sistema.
 - ◆ Reducir velocidad.
 - ◆ Prohibir nuevas conexiones.
 - ◆ Tirar paquetes.

Técnicas de Control de Congestión

- ◆ Otra clasificación puede ser:
 - Contrapresión
 - Paquetes de obstrucción
 - Señalización implícita de congestión
 - Señalización explícita de congestión

Contrapresión

- ◆ Técnica punto a punto
- ◆ Se puede propagar hacia atrás
- ◆ Se puede utilizar a nivel de enlace o de conexiones lógicas:
 - Contrapresión en conexiones lógicas con mucho tráfico, sin afectar a las de menor carga.

Paquetes de obstrucción

- ◆ Paquete generado por un nodo congestionado hacia un nodo origen.
- ◆ Ejemplo:
 - Paquete Ralentización del Emisor ("Source Quench") usado en ICMP.
- ◆ Posible enviar paquete de obstrucción antes de llegar a la congestión.
- ◆ Es una técnica ineficiente.

Señalización Implícita

- ◆ El propio emisor detecta la posible congestión:
 - Aumenta el retardo de propagación.
 - Se rechazan paquetes.
- ◆ Responsabilidad de los sistemas finales y no de los nodos intermedios.

Señalización Explícita

- ◆ La red alerta a los sistemas finales de la congestión y éstos toman medidas para reducir la carga.
- ◆ Sentido de la señalización:
 - Hacia atrás
 - Hacia adelante
- ◆ Técnicas divididas en tres categorías:
 - Binarias
 - Basadas en crédito
 - Basadas en velocidad

Técnicas de Señalización Explícita

- ◆ Binarias
 - El nodo congestionado activa un bit en un paquete. El emisor disminuye su flujo de tráfico por la conexión lógica.
- ◆ Basadas en crédito
 - Cuando el emisor agota su crédito, debe esperar a que se le conceda más.
- ◆ Basadas en velocidad
 - El emisor tiene un límite en la velocidad de transmisión por una conexión lógica. Los nodos intermedios pueden enviar paquetes hacia el emisor para variar dicho límite.

Algoritmos de Control de Congestión

◆ Veremos:

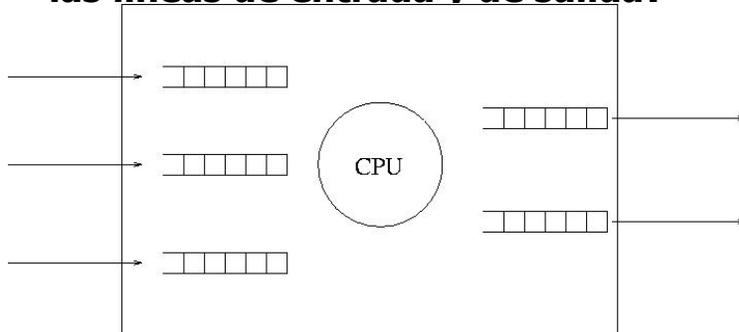
- Descarte de paquetes (bucle cerrado)
- Paquetes reguladores (bucle cerrado)
- Traffic Shapping (bucle abierto)

Descarte de paquetes

- ◆ Cuando un nodo tiene saturados sus recursos (buffers), se tiran paquetes:
 - Datagramas se pierden.
 - C.V. se retransmite.
- ◆ Problema 1: **si el paquete recibido es un ACK y se tira por no tener espacio para guardarlo, se origina una retransmisión.**
 - Solución: **reservar posiciones en el buffer para análisis de tráfico. Si es ACK se acepta y si no, se tira.**

Descarte de paquetes

◆ Problema 2: **¿Cómo se asignan buffers a las líneas de entrada y de salida?**



Descarte de paquetes

◆ Problema 2 (continuación) **se proponen tres tipos de técnica para realizar esa asignación:**

- Asignación dinámica (en base al uso): **No es eficiente, porque si una línea se carga, acapara todos los recursos (iniciación de las otras).**
- Asignación fija: **No es eficiente ya que podemos tener líneas con buffers vacíos y otras saturadas.**
- Subóptima: **Mezcla de las anteriores. Se reserva un número fijo de posiciones en el buffer para cada línea y el sobrante se asigna dinámicamente.**

Paquetes reguladores

- ◆ También conocidos como ***choke packet***.
- ◆ Los nodos monitorizan las líneas de salida, asociándoles un peso en función del uso:

$$U_n = aU_{n-1} + (1-a)f$$

$a, U_i \in [0,1]$

$$f = \begin{cases} 0 & \text{no se transmite actualmente} \\ 1 & \text{se está transmitiendo} \end{cases}$$

a permite dar mayor o menor importancia a la historia reciente.
Si $U > U_{\text{umbral}}$ la línea se pone en *alerta*.

Paquetes reguladores

- ◆ Si se tiene que encaminar por una línea en alerta:
 - Se envía al origen un paquete regulador.
 - El paquete se encamina normalmente, activando un bit que informa a los siguientes nodos que el origen está avisado.
- ◆ Recibido el aviso, el origen:
 - Disminuye el tráfico.
 - Pasado un tiempo sin recibir paquetes de regulación, se vuelve a subir la tasa.

Paquetes reguladores

◆ Variaciones:

- Mandar paquetes reguladores con información de estado (grave, muy grave, etc.)
- Monitorizar también el tamaño de las colas.
- Pedirle al nodo anterior, que encamine por otro nodo.

Traffic Shapping

◆ **Objetivo: adecuar el tráfico de salida de un host con independencia de los patrones de tráfico generado, evitando ráfagas. Se trata de mantener el tráfico constante, en definitiva, regular la tasa media de transmisión.**

◆ Relación con protocolos de ventana:

- El protocolo de ventana limita el número de paquetes en tránsito, pero no la velocidad con la que se ponen en la red.
- El traffic shapping regula la tasa a la que la información es enviada a la red.

Traffic Shapping

◆ **IMPORTANTE: se requiere un acuerdo entre el usuario y el proveedor de red.**

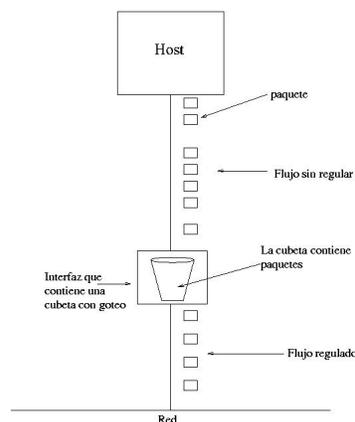
- Si el tráfico inyectado se adecúa al perfil pactado, el proveedor cursa dicho tráfico por la red. De otra forma, el tráfico se tira.

◆ **Ejemplos:**

- Leaky Bucket
- Token Bucket

Leaky Bucket

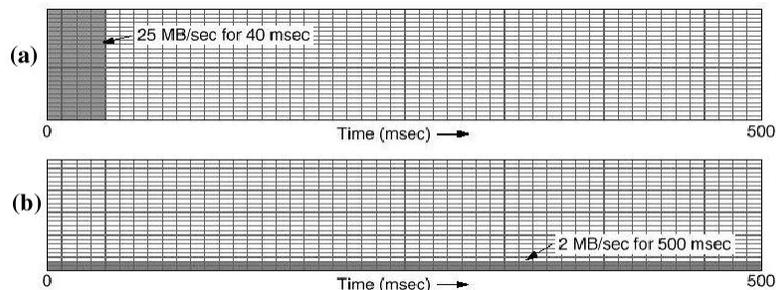
◆ **Este mecanismo convierte un flujo desigual de paquetes de un host, en un flujo continuo de paquetes hacia la red, moderando las ráfagas.**



Leaky Bucket

◆ Ejemplo

- (a) – Salida del host
- (b) – Salida del bucket



Leaky Bucket

◆ Implementación:

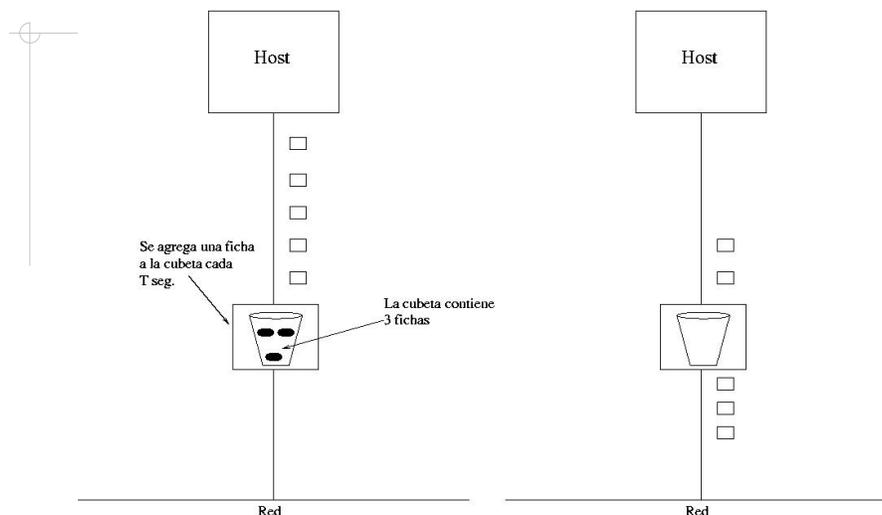
- El leaky bucket consiste en una cola finita.
- Al llegar un paquete, si hay espacio, se almacena. En caso contrario, se descarta.
- En cada pulso de reloj, se transmite un paquete (si existe)

◆ Usado en redes ATM.

Token Bucket

- ◆ Leaky Bucket impone un patrón de salida rígid → tasa promedio.
- ◆ En token bucket se permite picos de tráfico durante un pequeño intervalo.
- ◆ Funcionamiento:
 - La cubeta (bucket) contiene fichas (tokens).
 - Las fichas se insertan en la cubeta cada T seg.
 - Para transmitir, el emisor debe consumir una ficha.
 - Si no existe ficha, se espera.

Token Bucket



Token Bucket

◆ Parámetros importantes:

- **C capacidad de la cubeta (MB/s)**
- $\rho = \frac{1}{T}$ **tasa de creación de tokens (Bytes/s)**
- **M capacidad de la línea (Bytes)**
- **S duración de la ráfaga (segundos)**

$$C + \rho \times S = M \times S$$

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

Token Bucket

$$M = 25 \text{ MB/s}$$

$$\rho = 2 \text{ MB/s}$$

