
Temas 15 y 16

Redes de área local

Eytan Modiano

CSMA (*Carrier Sense Multiple Access*)

- En determinadas situaciones, los nodos se pueden oír entre ellos escuchando el canal: “Carrier Sensing”
- CSMA: versión formal de Aloha
 - Los nodos escuchan el canal antes de empezar a transmitir
 - Canal vacío => Transmiten
 - Canal ocupado => Esperan (van acumulando)
 - ¿Cuándo transmiten los nodos que están a la espera?

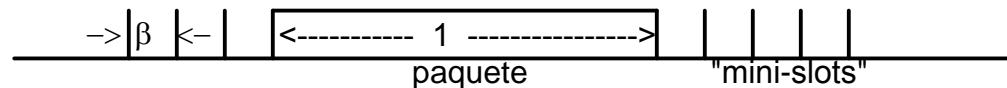
Cuando el canal vuelve a estar vacío, los nodos en espera intentan transmitir con probabilidad $q_r = 1$

Protocolo persistente, $q_r = 1$

Protocolo no persistente, $q_r < 1$

CSMA

- Sea τ = el retardo máximo de propagación en el canal:
 - Cuando un nodo inicia o detiene la transmisión, a todos los nodos les llevará este tiempo detectar si el canal está ocupado o vacío
- Para una comprensión básica, observar el sistema dividido en "mini-slots" de duración igual al retardo máximo de propagación
 - Normalizar la duración del "mini-slot" como $\beta = \tau/D_{tp}$ y establecer la duración del paquete = 1



- Los sistemas reales no están divididos en *slots*, pero este sistema hipotético simplifica el análisis y la comprensión del CSMA

Reglas del CSMA con *slots*

- **Cuando llega un nuevo paquete:**
 - Si el actual *mini-slot* está vacío, se empieza a transmitir en el siguiente *mini-slot*
 - Si el actual *mini-slot* está ocupado, el nodo permanece a la espera, acumulando
 - Si se produce una colisión, los nodos que se ven envueltos en la misma pasan a posición de espera
- **Los nodos en espera intentan transmitir tras un *mini-slot* vacío con probabilidad $q_r < 1$ (no persistente):**
 - Los intentos de transmisión sólo se producen tras un *mini-slot* vacío
 - Cada "período ocupado" (éxito o colisión) debe ir seguido de un *slot* vacío antes de que pueda iniciarse una nueva transmisión
- **El tiempo se puede dividir en épocas:**
 - Un paquete con éxito seguido de un *mini-slot* vacío (duración = $\beta+1$)
 - Una colisión seguida de un *mini-slot* vacío (duración = $\beta+1$)
 - Un *mini-slot* vacío (duración = β)

Análisis del CSMA

- Sea el estado del sistema el número de nodos en espera
- Sean los tiempos de transición de estado el final de los *slots* vacíos:
 - $T(n)$ = Promedio de tiempo entre las transiciones de estados cuando el sistema está en el estado n

$$T(n) = \beta + (1 - e^{-\lambda\beta} (1-q_r)^n)$$

Cuando q_r es pequeño $(1-q_r)^n \sim e^{-q_r n} \Rightarrow T(n) = \beta + (1 - e^{-\lambda\beta - nq_r})$

- Al comienzo de cada época, cada nodo en espera transmite con probabilidad q_r
- Se transmiten también las nuevas llegadas que se hayan producido durante el anterior *slot* vacío
- Con acumulación en espera n , el número de paquetes que se intentan transmitir al comienzo de una época es casi de Poisson con tasa:

$$g(n) = \lambda\beta + nq_r$$

Análisis del CSMA

- La probabilidad de éxito (por época) es:

$$P_s = g(n) e^{-g(n)}$$

- La duración esperada de una época es de aproximadamente:

$$T(n) \sim \beta + (1 - e^{-g(n)})$$

- De este modo, la tasa de éxito por unidad de tiempo es:

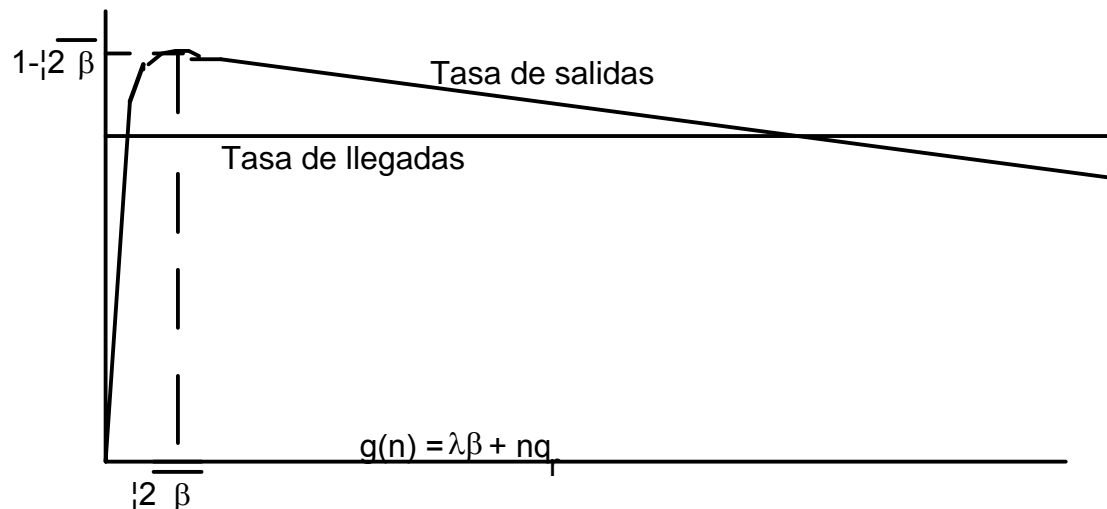
$$\lambda < \text{tasa de salidas} = \frac{g(n)e^{-g(n)}}{\beta + 1 - e^{-g(n)}}$$

Tasa máxima de transferencia del CSMA

- El valor óptimo de $g(n)$ se puede obtener:

$$g(n) \approx \sqrt{2\beta} \quad \lambda < \frac{1}{1 + \sqrt{2\beta}}$$

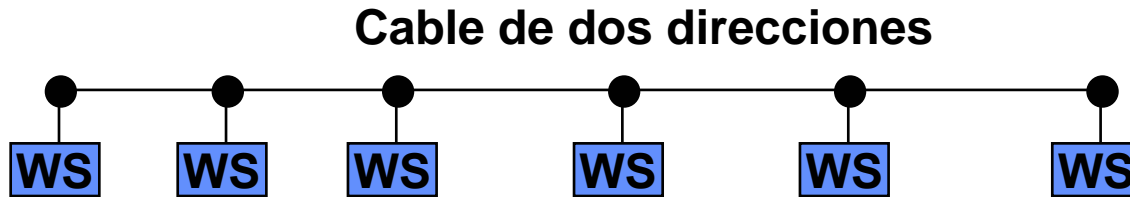
- Intercambio entre *slots* vacíos y tiempo perdido en las colisiones
- Tasa de transferencia elevada cuando β es pequeño
- Problemas de estabilidad similares a los de Aloha (menos críticos)



CSMA sin *slots*

- **El CSMA con *slots* no es práctico:**
 - Resulta difícil mantener la sincronización
 - Los *mini-slots* son útiles para entenderlo, pero no son necesarios para el funcionamiento del CSMA
- **El CSMA sin *slots* tendrá una tasa de transferencia ligeramente inferior debido al aumento de las probabilidades de colisión**
- **El CSMA sin *slots* tiene un valor eficaz β menor que el CSMA con *slots*:**
 - Básicamente, β pasa a ser el retardo medio de propagación, en lugar del retardo máximo

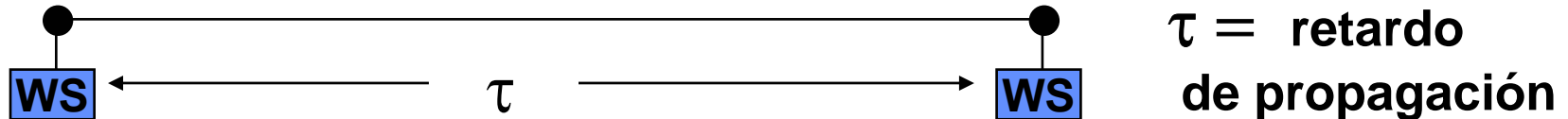
CSMA/CD y *Ethernet*



- **CSMA con capacidad de Detección de colisiones (CD):**
 - Los nodos son capaces de detectar colisiones
 - Al detectar una colisión, los nodos detienen la transmisión:
Esto reduce la cantidad de tiempo que se pierde en las colisiones

- **Protocolo:**
 - Todos los nodos escuchan las transmisiones del canal
 - Cuando un nodo tiene un paquete para enviar:
Canal vacío => Transmite
Canal ocupado => espera un tiempo aleatorio (retardo exponencial binario)
 - Si un nodo que está transmitiendo detecta una colisión, detiene la transmisión:
Espera un tiempo de retardo aleatorio y lo intenta otra vez

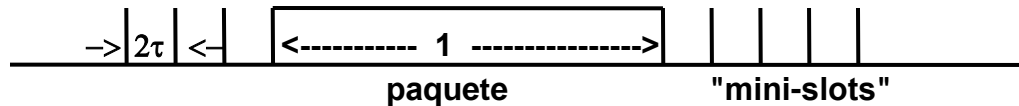
Tiempo para detectar las colisiones



- Se puede producir una colisión mientras la señal se propaga entre los dos nodos
- Ambos usuarios tardarán un tiempo adicional (retardo de propagación) hasta detectar la colisión y detener la transmisión
- Si τ es el retardo máximo de propagación en el cable, en caso de producirse una colisión, a todos los nodos implicados en la misma les puede llevar hasta 2τ segundos el detectar la colisión y detener la transmisión

Modelo aproximado de CSMA/CD

- Aproximación simplificada para una mejor comprensión
- Supongamos un sistema dividido en “mini-slots” de duración 2τ



- Si un nodo empieza a transmitir al comienzo de un *mini-slot*, al final del *mini-slot*, o bien:
 - No se ha producido ninguna colisión y el resto de las transmisiones no se verán interrumpidas
- O bien:
 - Se ha producido una colisión, pero al final del *mini-slot* el canal ya está vacío otra vez
- De este modo, una colisión afecta como mucho a un *mini-slot*

Análisis de CSMA/CD

- Supongamos que hay N usuarios y que cada uno de ellos intenta transmitir durante un “mini-slot” vacío con probabilidad p :
 - P incluye las nuevas llegadas y las retransmisiones

$$P(\text{intento de } i \text{ usuarios}) = \binom{N}{i} P^i (1-P)^{N-i}$$

$$P(\text{exactamente 1 intento}) = P(\text{éxito}) = NP(1-P)^{N-1}$$

Para maximizar $P(\text{éxito})$,

$$\frac{d}{dp} [NP(1-P)^{N-1}] = N(1-P)^{N-1} - N(N-1)P(1-P)^{N-2} = 0$$

$$\Rightarrow P_{op} = \frac{1}{N}$$

⇒ Tasa media de intentos de uno por *slot*

⇒ Obsérvese la similitud que hay con el Aloha con *slots*

Análisis de CSMA/CD (continuación)

$$P(\text{éxito}) = NP(1-p)^{N-1} = \left(1 - \frac{1}{N}\right)^{N-1}$$

$$P_s = \lim_{N \rightarrow \infty} P(\text{éxito}) = \frac{1}{e}$$

X = Promedio de *slots* por transmisión con éxito

$$P(X=i) = (1-P_s)^{i-1}P_s$$

$$\Rightarrow E[X] = \frac{1}{P_s} = e$$

- Una vez que se ha capturado con éxito un *mini-slot*, la transmisión continúa sin interrupción
- En el siguiente *mini-slot* se iniciarán nuevos intentos de transmisión, una vez que haya terminado la transmisión actual del paquete

Análisis de CSMA/CD (continuación)

- Sea S = el promedio de tiempo entre las transmisiones de paquetes que tienen éxito:

$$S = (e-1)2\tau + D_{Tp} + \tau$$

Mini-slots
vacíos/con colisión

Tiempo de transmisión de paquetes

Promedio de tiempo hasta el inicio del siguiente *mini-slot*

- Eficacia = $D_{Tp}/S = D_{Tp} / (D_{Tp} + \tau + 2\tau(e-1))$
- Sea $\beta = \tau / D_{Tp} \Rightarrow$ Eficacia $\approx 1/(1+4.4\beta) = \lambda < 1/(1+4.4\beta)$

- Comparar con un CSMA sin CD donde: $\lambda < \frac{1}{1+\sqrt{2\beta}}$

Notas sobre el CSMA/CD

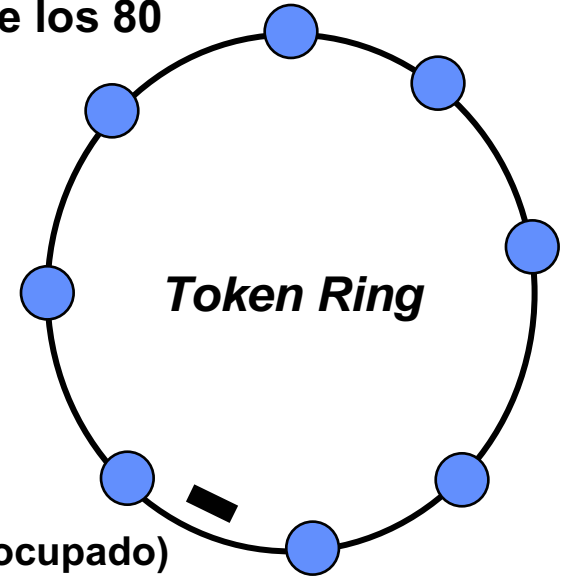
- Se puede ver como un sistema de reservas en el que se emplean los *mini-slots* para hacer las reservas para los *slots* de datos
- En este caso, Aloha se utiliza para hacer las reservas durante los *mini-slots*
- Una vez que un usuario captura un *mini-slot*, continúa transmitiendo sin interrupciones
- Por supuesto, en la práctica, no hay *mini-slots*:
 - Supone un impacto mínimo sobre el rendimiento, pero el análisis es más complejo

Ejemplos de CSMA/CD

- **Ejemplo (*Ethernet*):**
 - Tasa de transmisión = 10 Mbps
 - Tamaño de los paquetes = 1000 bits, $D_{Tp} = 10^{-4}$ seg
 - Distancia del cable = 1 milla, $\tau = 5 \times 10^{-6}$ seg
 - ➔ $\beta = 5 \times 10^{-2}$ y $E = 80\%$
- **Ejemplo (satélite GEO) - 1/4 de segundo de retardo de propagación**
 - $\beta = 2,500$ y $E \sim 0\%$
- **CSMA/CD sólo es válido en casos de propagación corta**
- **¿Cómo se expande *Ethernet* a 100 Mbps?**
- **¿Cómo se expande *Ethernet* a 1 Gbps?**

Redes en anillo (*Token rings*)

- Las redes en anillo las desarrolló IBM a comienzos de los 80
- El testigo (*token*): una secuencia de *bit*:
 - El testigo circula por el anillo:
Testigo ocupado: 01111111
Testigo libre: 01111110
- Cuando un nodo quiere transmitir:
 - Espera a que haya un testigo libre
 - Retira el testigo del anillo (lo sustituye por un testigo ocupado)
 - Transmite el mensaje
 - Cuando acaba de transmitir, vuelve a introducir el testigo libre en el anillo
 - Los nodos deben almacenar 1 *bit* de datos, de modo que se puede cambiar un testigo libre por un testigo ocupado
- Las redes *Token ring* son básicamente un sistema de sondeo
El testigo realiza el sondeo



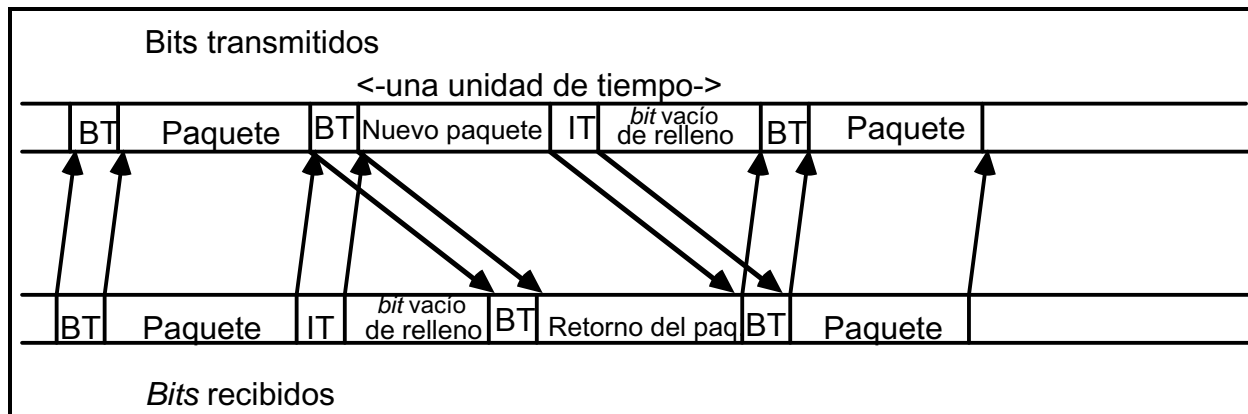
Puesta en libertad del testigo (*token*)

- **Puesta en libertad tras la transmisión:**
 - El nodo vuelve a introducir el testigo en el anillo tan pronto como finaliza la transmisión del paquete
 - El siguiente nodo puede utilizar el testigo tras un corto retardo de propagación
- **Puesta en libertad tras la recepción:**
 - El nodo sólo libera el testigo una vez que su propio paquete vuelve a él:
Sirve simplemente como un mecanismo de acuse de recibo

TRANSMISIÓN DE PAQUETES

(puesta en libertad tras la transmisión)

- Cuando no transmiten sus propios paquetes, los nodos retransmiten todo lo que reciben
- Tras recibir un testigo vacío, un nodo puede empezar a enviar un nuevo paquete (y rechazar los *bits* entrantes)
- Una vez que un nodo envía un paquete y el testigo vacío, envía un *bit* vacío de relleno hasta que:
 - El paquete seguido de un testigo vacío o ocupado, vuelva al anillo



Análisis de retardos

- Se puede analizar el sistema mediante los resultados de las reservas multiusuario
- Sistema exhaustivo - Los nodos vacían sus colas antes de pasar el testigo al siguiente nodo
- Supongamos m nodos, cada uno de ellos con llegadas de Poisson de tasa λ/m
- Sea v = el retardo medio de transmisión del testigo y propagación
- Se puede ver como un sistema de reservas con m usuarios y un intervalo medio de reservas (ver los resultados del sistema de reservas)

$$W = \frac{\lambda E[X^2] + v(m - \rho)}{2(1 - \rho)}, \quad \rho = m(\lambda / m)E[X] = \lambda E[X]$$

- Obsérvese que en un sistema exhaustivo se puede lograr una tasa de transferencia del 100%

Análisis de la tasa de transferencia (no exhaustivo)

- ***Gated system* con servicio limitado: cada nodo está limitado a enviar un solo paquete cada vez:**
 - Cuando el sistema está muy cargado, los nodos están siempre ocupados y tienen un paquete para enviar
- **Supongamos que cada nodo transmite un paquete y, a continuación, pasa el testigo al siguiente nodo:**
 - V_i = tiempo de transmisión y propagación del testigo entre dos nodos (el tiempo de transmisión suele ser despreciable)
- **Cantidad de tiempo que se tarda en transmitir N paquetes:**

$$T_N = N \cdot E[X] + V_1 + V_2 + \dots + V_N = N \cdot E[X] + N \cdot E[V]$$

$$\lambda < N \cdot E[X] / (N \cdot E[X] + N \cdot E[V]) = 1 / (1 + E[V] / E[X])$$

- **Compárelo con el CSMA/CD, pero observe que V es el retardo entre dos nodos y no el retardo máximo en el cable**

Análisis de la tasa de transferencia (puesta en libertad del testigo tras la recepción)

- Los nodos sólo liberan el testigo después de que haya vuelto
- Una vez más, suponer que cada nodo envía un paquete cada vez
- Tiempo total para enviar UN paquete:
- $T = E[X] + V_1 + V_2 + \dots + V_m + V_i$
 - ← Tiempo para enviar el testigo al siguiente nodo
 - ← Hay M nodos en el anillo
- $T = E[X] + (m+1)E[V] \Rightarrow$

$$\lambda < E[X]/T = 1/(1+(m+1)E[V]/E[X])$$

Análisis del retardo

- **Puesta en libertad tras la transmisión:**
 - *Partially gated system* con servicio limitado (apt. 3.5.2)

$$W = \frac{\lambda E[X^2] + v(m + \lambda E[X])}{2(1 - \lambda E[X] - \lambda v)}$$

- **Puesta en libertad tras la recepción:**
 - Problema 4.27 para realizar en casa
 - Al tiempo de transmisión del paquete se puede añadir un tiempo adicional equivalente a una vuelta del recorrido

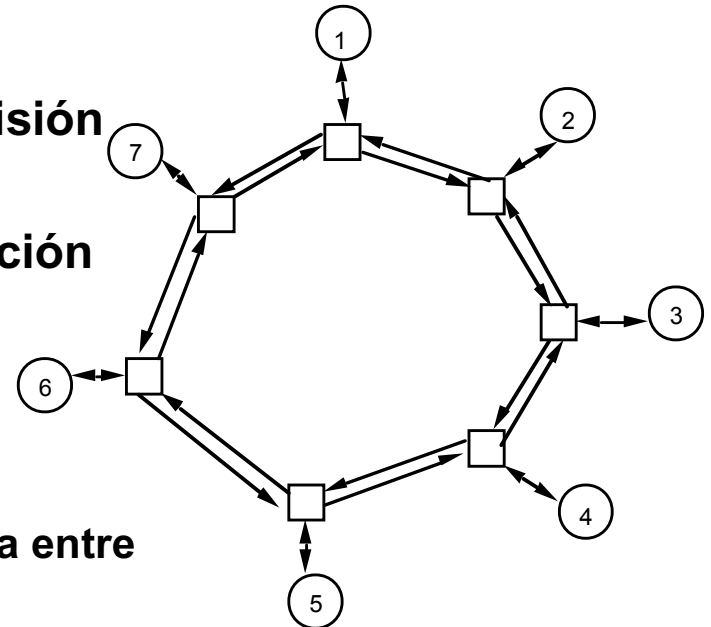
$$W = \frac{\lambda(E[X^2] + 2mv + m^2v^2) + v(m + \lambda(E[X] + mv))}{2(1 - \lambda(E[X] + (m + 1)v))}$$

Problemas de las redes *Token ring*

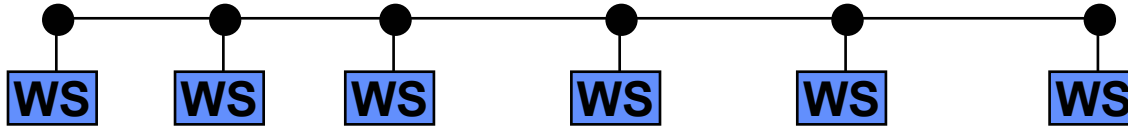
- **Reparto equitativo: ¿puede un nodo conservar el testigo durante un largo período de tiempo?**
 - **Solución:** establecer un tiempo máximo para la conservación del testigo
- **Fallo de testigos: los testigos se pueden crear o destruir debido al ruido:**
 - **Solución distribuida:**
 - Los nodos pueden reconocer la pérdida de un testigo y crear uno nuevo
 - Las colisiones se producen cuando dos o más nodos crean un nuevo testigo a la vez => son necesarios los algoritmos de resolución de colisiones
- **Fallos en los nodos: puesto que cada uno de los nodos debe retransmitir todos los datos entrantes, un fallo en cualquiera de ellos afectará al funcionamiento de todo el anillo**
- **Estándar de *Token ring*: IEEE 802.5**

FDDI

- La interfaz de datos distribuida por fibra (FDDI) es un estándar de LAN de tipo *Token Ring* en fibra óptica de 100Mbps
- El FDDI consta de dos anillos cuyo tráfico viaja en direcciones opuestas:
 - Los fallos sencillos se pueden aislar pasando de un anillo al otro a cada lado del fallo
- El testigo se libera después de la transmisión
- Hay un límite de tiempo para la conservación del testigo
- Hay un límite (superior) de tiempo entre las visitas de un testigo a un nodo:
 - Permite los retardos garantizados
 - Limita el tamaño de un anillo (la distancia entre los nodos y el número de nodos)
- El FDDI se diseñó como tecnología para una MAN o una CAN



BUSES DE TESTIGOS



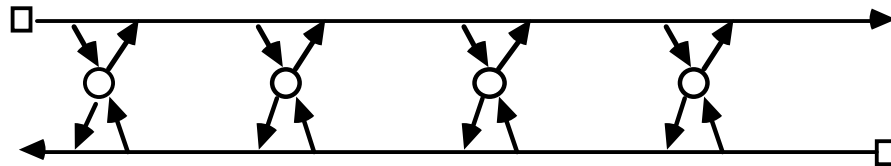
- Un paquete de control espacial actúa como testigo
- Los nodos han de estar en posesión del testigo para transmitir
- El testigo se pasa de un nodo a otro siguiendo un orden:
 - Desde el punto de vista conceptual, un bus de testigos es lo mismo que una *Token ring*
 - Cuando un nodo termina la transmisión, envía un testigo vacío al siguiente nodo (dirigiendo adecuadamente el paquete de control)
 - Similar a un sistema de sondeo
- Problemas:
 - La eficacia es menor que la de las *Token rings* debido a que tanto el retardo de transmisión de los paquetes como los retardos de propagación son superiores
 - Es necesario un protocolo para entrar y salir del bus

TESTIGOS IMPLÍCITOS

- En un *token bus*, los testigos vacíos se pueden sustituir por silencios
- El siguiente nodo empieza a transmitir un paquete cuando observa que el bus está en silencio
- Si el siguiente nodo no tiene ningún paquete, empezará el siguiente y así sucesivamente con un retardo cada vez mayor
- Si el retardo de propagación del bus es mucho menor que el tiempo de transmisión de un testigo, esto puede reducir el retardo
- Este esquema se utiliza para las redes LAN inalámbricas (IEEE 802.11) y se conoce como CSMA/CA (evita colisiones)

BUS DUAL DE COLA DISTRIBUIDA (DQDB)

- Red MAN con dos buses unidireccionales de 150 Mbps cuyo tráfico viaja en direcciones opuestas
- Todas las tramas tienen la misma longitud (53 bytes); las tramas vacías se generan en los extremos de inicio de los buses y las rellenan los nodos "al vuelo", a medida que pasan por ellos

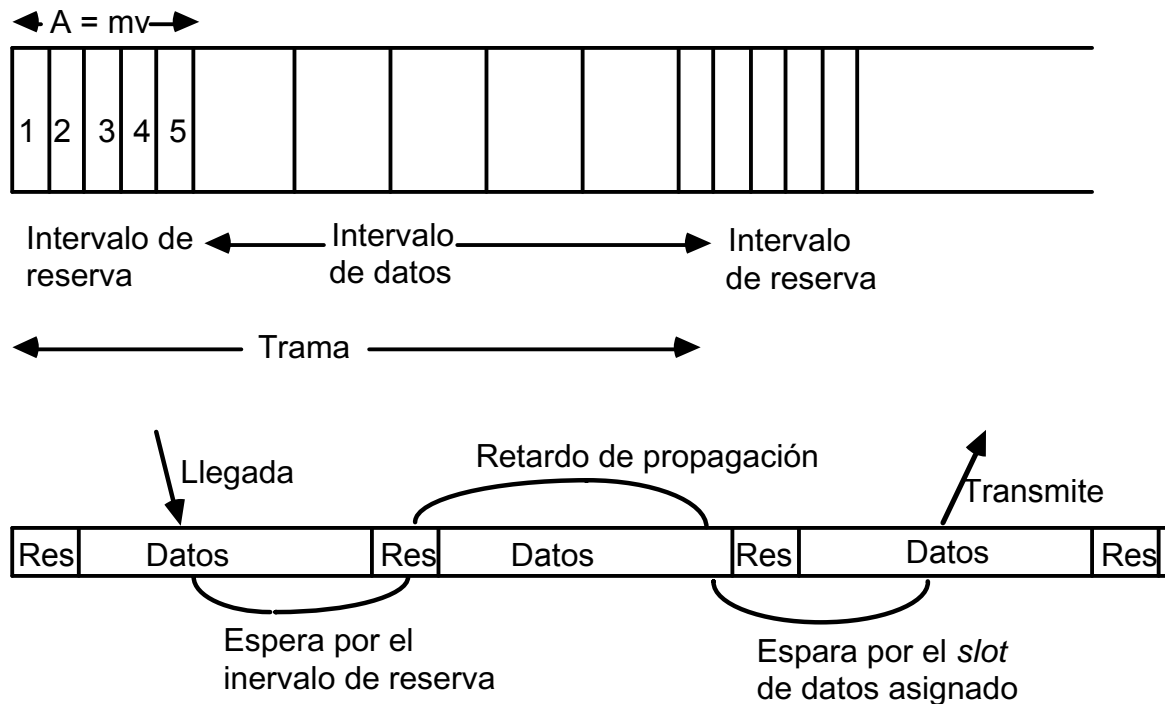


- Cada nodo utiliza el bus que se dirige a la derecha para enviar tramas a los nodos que están a su derecha y viceversa
- El DQDB se estandarizó como IEEE 802.6 y pretendía ser compatible con las ATM

Reservas en el DQDB

- El algoritmo *greedy*: cada nodo utiliza un *slot* libre cuando tiene que enviar algo:
 - De este modo, es posible una eficacia del 100%
- El problema de este enfoque trivial es que no resulta equitativo: los nodos que están en la cola del bus pueden esperar eternamente
- El DQDB utiliza sistemas de reservas mediante los cuales los nodos envían, en dirección contraria, peticiones para reservar los *slots* vacíos
 - Si un nodo tiene una trama para enviar por el bus de la derecha, introduce el *bit* de petición en una trama que envía por el bus de la izquierda
 - Los nodos mantienen una cola de peticiones “implícita” en la que el primero en llegar es el primero en ser servido (FCFS); de ahí que reciba el nombre de "cola distribuida"

Retardo de propagación grande (redes de satélites)



- **Sistema de reservas de satélites:**
 - Emplea los *mini-slots* para hacer las reservas para *slots* de datos mayores
 - El acceso a los *mini-slots* puede no ser eficaz (Aloha, TDMA, etc.)
- A grosso modo, el retardo es $3/2$ veces el retardo de propagación más el retardo de cola ideal

Reservas de satélites

- **La longitud de la trama debe exceder el retardo equivalente a una vuelta:**
 - Los *slots* de reserva durante la trama j se utilizan para reservar *slots* de datos en la trama $j+1$
 - Longitud variable: se sirven todas las peticiones de la trama j en la $j+1$
 - Es difícil mantener la sincronización
 - Es difícil ofrecer QoS (ej.: dar soporte al tráfico de voz)
 - Longitud fija: se mantiene una cola virtual de peticiones
- **Mecanismo de reserva:**
 - Controlador a bordo por satélite
 - Controlador en tierra
 - Algoritmo de colas distribuidas:
 - Todos los nodos monitorizan las peticiones de reserva y utilizan el mismo algoritmo para realizar las reservas
- **Control del acceso al canal:**
 - TDMA: es simple, pero resulta difícil añadir más usuarios
 - Aloha: puede dar soporte a un gran número de usuarios, pero es difícil solucionar las colisiones y añade un enorme retardo

Reservas con Aloha

- Se utiliza Aloha para capturar un *slot*
- Tras capturar un *slot* el usuario lo mantiene en su poder hasta el final de la operación:
 - Los otros usuarios observan que el *slot* está ocupado y no realizan ningún intento
- Una vez finalizada la operación, los otros usuarios pueden ir a por el *slot*
 - Los otros usuarios observan que el *slot* está vacío y realizan un intento utilizando Aloha
- Método útil para grandes transferencias de datos o para una mezcla de datos y voz:

Slot	1	2	3	4	5	6	
	15	vacío	3	20		2	trama 1
	15	7	3	vacío	9	2	trama 2
	vacío	7	3		9	vacío	trama 3
	18	7	3		9	6	trama 4
	18	7	3	15	9	6	trama 5

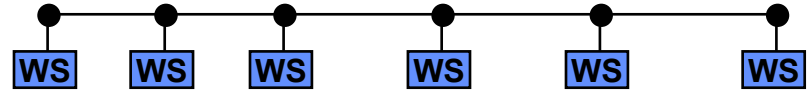
Acceso múltiple de paquetes. Resumen

- **Latencia: relación entre el retardo de propagación y el tiempo de transmisión de paquetes**
 - Ejemplo de GEO: $D_p = 0,5$ seg, longitud de paquetes = 1000 *bits*, $R = 1$ Mbps
Latencia = 500 => muy alto
 - Ejemplo de LEO: $D_p = 0.1$ seg
Latencia = 100 => aún muy alto
 - En los canales de satélites la tasa de datos debe ser muy baja para estar en un entorno de latencia bajo
- **Protocolos de latencia baja:**
 - CSMA, *Polling*, *Token Rings*, etc.
 - Tasa de transferencia $\sim 1/(1+a\alpha)$, α = latencia, a = constante
- **Protocolos de latencia alta:**
 - Aloha no es sensible a la latencia, pero generalmente presenta una tasa de transferencia baja:
Retardos muy pequeños
 - El sistema de reservas puede lograr una tasa de transferencia muy elevada:
Retardos para realizar las reservas
 - Los protocolos se pueden diseñar para que actúen como un híbrido entre Aloha y las reservas:
Aloha cuando la carga es baja y reservas cuando es elevada

Migración a redes LAN conmutadas

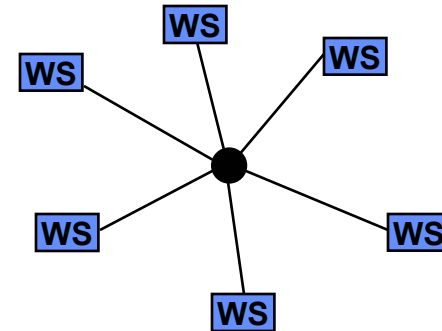
- **Ethernet tradicional:**

- **Nodos conectados con cable coaxial:**
Largas "líneas" de cable por todas partes
- **Protocolo CSMA/CD**



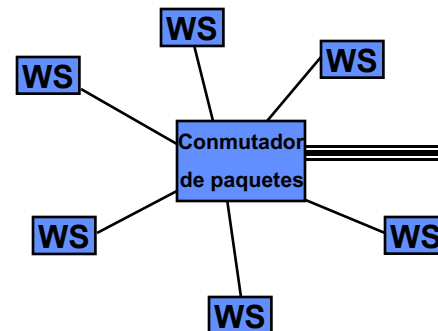
- **Ethernet con "Hub":**

- **Los nodos están conectados al hub**
El hub actúa como un repetidor de difusión
Cable acortado, útil para 100 Mbps
- **Protocolo CSMA/CD**
- **Es fácil añadir/quitar usuarios**
- **Es fácil localizar errores**
- **Cableado barato (par trenzado, 10baseT)**



- **Ethernet con conmutador (switch):**

- **Sin CSMA/CD:**
Es fácil aumentar la tasa de datos (ej.: Gbit Ethernet)
- **Los nodos transmiten cuando quieren**
- **El conmutador almacena en cola los paquetes y los transmite a su destino**
- **Capacidad habitual de un conmutador: 20-40 puertos**
- **Cada nodo puede transmitir ahora a la tasa total de 10/100/Gbps**
- **Modularidad: Los conmutadores pueden estar conectados entre sí a través de puertos de alta velocidad**



Conexión a otros conmutadores