

Módulo 05

La Capa de Enlace

(Pt. 2)



Redes de Computadoras
Depto. de Cs. e Ing. de la Comp.
Universidad Nacional del Sur



Copyright

- Copyright © 2010-2024 A. G. Stankevicius
- Se asegura la libertad para copiar, distribuir y modificar este documento de acuerdo a los términos de la **GNU Free Documentation License**, versión 1.2 o cualquiera posterior publicada por la Free Software Foundation, sin secciones invariantes ni textos de cubierta delantera o trasera
- Una copia de esta licencia está siempre disponible en la página <http://www.gnu.org/copyleft/fdl.html>
- La versión transparente de este documento puede ser obtenida de la siguiente dirección:

<http://cs.uns.edu.ar/~ags/teaching>

Contenidos

- Servicios provistos por la capa de enlace
- Protocolos de acceso múltiple
- Direcciones de red local y protocolo **ARP**
- Ethernet
- Hubs, bridges y switches
- Enlaces inalámbricos
- Virtualización de enlaces
- Datacenters

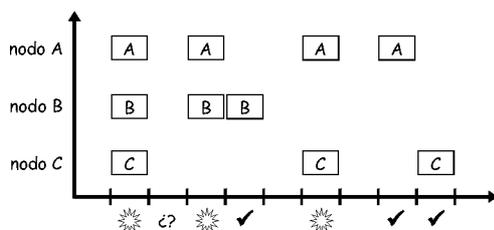
ALOHA con slots

- **ALOHA** con slots puede ser aplicado cuando se cumplan las siguientes condiciones:
 - Todos los frames son del mismo tamaño
 - El tiempo se divide en slots de igual tamaño, justo la cantidad necesaria para transmitir un frame
 - Los nodos sólo comienzan a transmitir frames al comienzo de un slot
 - Los nodos deben estar sincronizados
 - Si dos o más nodos eligen transmitir en el mismo slot, todos los nodos son capaces de detectar la colisión

ALOHA con slots

- De satisfacerse estas condiciones, la operatoria de **ALOHA** con slots es bastante sencilla:
 - Para enviar un frame, un determinado nodo debe esperar al comienzo del siguiente slot para transmitirlo en ese momento
 - De no producirse una colisión, el nodo puede enviar el próximo frame en el siguiente slot
 - En cambio, si se produce una colisión, el nodo en cuestión intentará retransmitir el frame en los slots subsiguientes con una probabilidad p hasta que lo logre enviar correctamente

ALOHA con slots



Análisis

● Aspectos positivos:

- Un nodo cuando es el único emitiendo puede transmitir de manera continua haciendo uso de la totalidad del ancho de banda del canal
- Es un protocolo altamente descentralizado, los nodos sólo deben ponerse de acuerdo cuándo comienza y cuándo terminan los slots
- Simple

Análisis

● Aspectos criticables:

- Es un protocolo que contempla colisiones, esto es, contempla desperdiciar durante un slot la totalidad del ancho de banda del canal
- El mecanismo de recuperación ante colisiones puede desperdiciar slots (cuando el nodo elige no transmitir en un slot donde iba a ser el único transmitiendo)
- Los nodos quizás puedan detectar que se produjo una colisión en menos tiempo de lo que toma transmitir la totalidad de la trama

Eficacia del protocolo

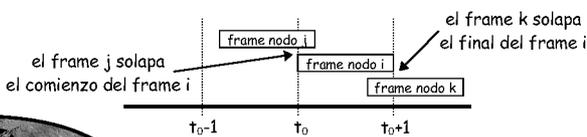
- Definiremos como eficacia del protocolo a la fracción de slots útiles a lo largo del tiempo cuando múltiples nodos desean varios frames
- Supongamos que existen **N** nodos, donde múltiples nodos desean transmitir y cada uno intenta hacerlo con una probabilidad **p**
 - Probabilidad de que el primer nodo tenga éxito en un determinado slot: $p(1-p)^{N-1}$
 - Probabilidad de que todos los nodos tengan éxito en su transmisión: $Np(1-p)^{N-1}$

Eficacia del protocolo

- ¿Para qué valor de p se alcanza la máxima eficacia con N nodos?
 - Para alcanzar la máxima eficacia para N nodos debemos encontrar el p^* que maximice $Np^*(1-p^*)^{N-1}$
 - Para generalizar este resultado a múltiples nodos, podemos tomar el límite de $N(1/N)(1-(1/N))^{N-1}$ cuando N tiende hacia $+\infty$, lo que da $1/e = 0.37$
- Esto es, en el mejor de los casos el canal termina siendo usado para transmitir información útil sólo el 37% del tiempo

ALOHA puro

- El protocolo **ALOHA** puro (esto es, sin slots), es un tanto más sencillo:
 - Los nodos no requieren sincronizarse entre sí
 - Si un nodo tiene que transmitir un frame, lo comienza a transmitir inmediatamente
 - Evidentemente, la probabilidad de que se produzca una colisión aumenta



¿Por qué es tan complicado?

¿por qué es tan complicado sincronizar un sistema distribuido?

Show drafts

La sincronización de un sistema distribuido es un problema complejo por varias razones:

Falta de un reloj global: En un sistema distribuido, cada nodo tiene su propio reloj local. Estos relojes no son perfectamente precisos y pueden variar entre sí, lo que dificulta determinar el orden en que ocurrieron los eventos.

Comunicación asíncrona: Los mensajes entre nodos se envían de forma asíncrona, lo que significa que pueden tardar un tiempo impredecible en llegar a su destino. Esto puede hacer que sea difícil saber cuándo se ha completado una operación y cuándo es seguro que otro nodo actúe en consecuencia.

Fallos y concurrencia: Los nodos pueden fallar y los mensajes pueden perderse o retrasarse. Esto puede hacer que sea difícil mantener un estado coherente en todo el sistema y evitar condiciones de carrera, en las que dos o más nodos intentan realizar la misma operación al mismo tiempo.

Escalabilidad: A medida que un sistema distribuido crece en tamaño, la complejidad de la sincronización aumenta rápidamente. Se vuelve más difícil mantener la precisión del reloj, la coherencia del estado y evitar las condiciones de carrera.

Existen diferentes mecanismos para sincronizar sistemas distribuidos, cada uno con sus propias ventajas y desventajas. Algunos de los mecanismos más comunes incluyen:

- **Algoritmos de sincronización de relojes:** Estos algoritmos intentan sincronizar los relojes de los nodos para que tengan una idea más precisa del tiempo global.
- **Mecanismos de bloqueo:** Los bloqueos se utilizan para evitar que dos o más nodos accedan simultáneamente a un recurso compartido.
- **Protocolos de compromiso:** Los protocolos de compromiso se utilizan para garantizar que todos los nodos acuerden un estado común.

La elección del mecanismo de sincronización adecuado depende de las necesidades específicas del sistema distribuido.

Aquí hay algunos recursos adicionales que pueden ser útiles:

- Sincronización en Sistemas Distribuidos (se quitó una URL, no válida)
- Sincronización de relojes en un sistema distribuido
- Sistemas Distribuidos (se quitó una URL, no válida)

Espero que esta información te ayude a comprender mejor por qué la sincronización de un sistema distribuido es tan complicada.

Eficacia del protocolo

- El mismo análisis de la eficacia de **ALOHA** con slots puede extrapolarse a **ALOHA** puro:

→ Para que un cierto nodo transmita exitosamente se deben dar tres condiciones: que transmita, que ningún otro nodo transmita en el slot anterior y que tampoco transmitan en el actual

→ En otras palabras:

$$p \times (1-p)^{N-1} \times (1-p)^{N-1} = p \times (1-p)^{2(N-1)}$$

→ Con un valor óptimo para **p** y analizando el límite para **N** tendiendo a $+\infty$ da $1/(2e) = \underline{0.18}$

MAC vía CSMA

- El protocolo de acceso al medio **CSMA** (Carrier Sense Multiple Access) consiste esencialmente en escuchar antes de transmitir

→ Si el canal se detecta ocioso, se comienza a transmitir inmediatamente

→ Si el canal se detecta ocupado, se demora momentáneamente la transmisión

- **CSMA** es el protocolo que usualmente usamos los humanos al conversar... salvo en el asado de redes al terminar el cuatrimestre

Colisiones bajo CSMA

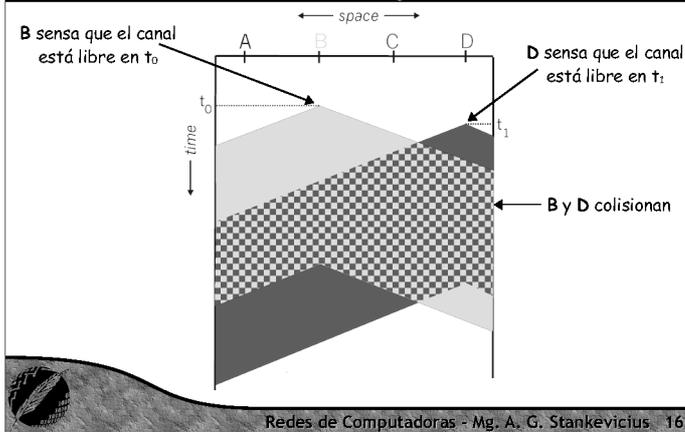
- A pesar de los recaudos que toma **CSMA**, las colisiones se siguen pudiendo producir

→ Los retardos en la propagación de las señales entre los nodos pueden causar que un nodo no escuche a tiempo que el canal está siendo usado

→ Al producirse una colisión, la totalidad de tiempo insumido en la transmisión de las tramas es desperdiciado

→ La distancia entre los nodos y el retardo de las señales determinan la probabilidad de colisión

Colisiones bajo CSMA



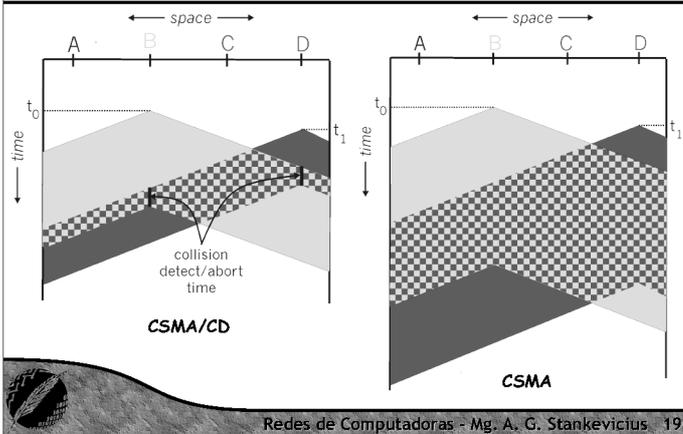
CSMA/CD

- La variante **CSMA/CD** (Collision Detect) se comporta de manera análoga a **CSMA**:
 - Se sensa el medio antes de transmitir y se posterga la transmisión en caso de estar siendo usado
 - La variante **CSMA/CD** requiere que las colisiones se detecten rápidamente
 - La idea es poder abortar lo antes posible aquellas transmisiones que entran en colisión, con el objeto de reducir el tiempo que se mantiene ocupado el canal con una transmisión que va a terminar siendo fallida

CSMA/CD

- Para implementar esta variante se debe poder detectar con rapidez las colisiones:
 - Esta variante se puede implementar fácilmente en las redes locales cableadas: basta con medir la fuerza de la señal y comparar lo que se transmite con lo que se recibe
 - En contraste, no es tan simple de implementar en los enlaces inalámbricos, pues la recepción queda inhibida al transmitir
- Este es el protocolo que adoptan los periodistas en la tele al dar notas vía satélite o por zoom

Colisiones en CSMA/CD



Protocolo MAC de ethernet

- Ethernet adopta a CSMA/CD como protocolo de acceso al medio compartido
 - Los adaptadores no transmiten si notan que otro adaptador está transmitiendo (esto es, antes de transmitir detecta si está presente la portadora)
 - Al transmitir el adaptador aborta ni bien descubre que otro adaptador comienza a transmitir (esto es, implementa la detección temprana de colisiones)
 - Antes de retransmitir el adaptador espera una cantidad aleatoria de tiempo (esto es, se trata de un protocolo de acceso aleatorio al medio)

Algoritmo CSMA/CD

- Algoritmo CSMA/CD:
 - **Paso 1)** El adaptador recibe el datagrama a ser enviado y lo encapsula en una trama
 - **Paso 2a)** Si el adaptador observa que el canal está disponible, comienza a transmitir la trama
 - **Paso 2b)** Si en cambio detecta que el canal está ocupado, espera hasta que se desocupe y reintenta
 - **Paso 3a)** Si el adaptador transmite todo el frame sin cruzarse con otro adaptador que este intentando transmitir al mismo tiempo, considera al frame correctamente transmitido

Algoritmo CSMA/CD

- Continúa:

- **Paso 3b1)** Si en cambio nota que otro adaptador intentó transmitir al mismo tiempo, aborta su transmisión y envía la señal de colisión (jam signal)
- **Paso 3b2)** Luego de abortar, el adaptador entra en la fase de espera exponencial: luego de la m -ésima colisión, elige un K al azar entre $\{0, 1, \dots, 2^m - 1\}$ y espera el tiempo necesario para transmitir $K \times 512$ bits para finalmente volver a **Paso 2)**

Algoritmo CSMA/CD

- Señal de colisión: es un patrón de 48 bits que se utiliza para que los todos adaptadores que están transmitiendo detecten la colisión

- Tiempo de transmisión de un bit:

- **0,1 μ s** para ethernet (10Mbps)
- **0,01 μ s** para fast ethernet (100Mbps)
- **0,001 μ s** para gigabit ethernet (1Gbps)
- Para $K = 1023$, el tiempo de espera es unos 50ms para ethernet 10Mbps, 5ms para fast ethernet y de 0,5ms para ethernet de 1Gbps

Algoritmo CSMA/CD

- Fase de espera exponencial:

- El objetivo es ajustar las retransmisiones de manera que estimen la carga actual
- Bajo una gran carga se estiran los tiempo de espera
- Primera colisión: se elige K de $\{0, 1\}$ (y se espera el tiempo de transmisión de $K \times 512$ bits)
- Segunda colisión: se elige K entre $\{0, 1, 2, 3\}$
- Luego de 10 colisiones: se elige K entre $\{0, 1, 2, \dots, 1021, 1022, 1023\}$

Eficacia de CSMA/CD

- No resulta sencillo extrapolar el análisis de la eficacia del protocolo **ALOHA** con y sin slots a esta versión del protocolo **CSMA/CD**
- Un tratamiento pormenorizado escapa al alcance del curso, sin embargo podemos repasar las conclusiones respecto a la eficacia
 - t_{prop} = tiempo máximo de propagación entre dos nodos de la LAN
 - t_{trans} = tiempo que toma transmitir una trama de tamaño máximo

Eficacia de CSMA/CD

- Bajo estos supuestos, la eficiencia de **CSMA/CD** se aproxima a la siguiente proporción:

$$\text{eficacia} = \frac{1}{1 + 5t_{prop} / t_{trans}}$$

- Obsérvese lo siguiente:
 - La eficacia se acerca al **100%** a medida que t_{prop} se acerca a **0**
 - También se acerca al **100%** a medida que t_{trans} se hace infinito

Eficacia de CSMA/CD

- En síntesis:
 - Brinda un comportamiento más eficaz que **ALOHA** con o sin slots
 - Retiene la naturaleza distribuida, sin depender de que previamente se sincronicen los relojes o siquiera el comienzo de las ranuras de transmisión
 - Es relativamente sencillo y por ende económico de implementar

MAC via "tomar turnos"

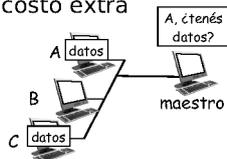
- Los protocolos de partición del canal se comportan usualmente de manera análoga:
 - Logran compartir el canal de manera eficiente y equitativa con mucha carga, cuando múltiples nodos desean usar el canal en simultáneo
 - Sin embargo, resultan poco efectivos con poca carga: se retrasa el acceso al canal, sólo $1 / N$ del ancho de banda es asignado a cada nodo, incluso cuando sea el único tratando de usar el canal

MAC via "tomar turnos"

- Los protocolos de acceso aleatorio también se comportan usualmente de manera análoga:
 - Son altamente eficientes cuando hay poca carga: de haber un único nodo activo, puede hacer uso de la totalidad del ancho de banda
 - No tan eficiente con mucha carga: aparece colisiones, las que redundan en desperdiciar ancho de banda
- Los protocolos de acceso al medio basados en "tomar turnos" intentan ofrecer lo mejor de estas dos alternativas

¿Cómo tomar turnos?

- Una posibilidad es tomar turnos indagando a cada uno de los participantes
 - Se distingue a un nodo como maestro, el cual indaga nodo a nodo invitándolos a hacer uso del canal en esa oportunidad
- Análisis:
 - El indagado (polling) insume un costo extra
 - La latencia se dispara
 - Se introduce un punto único de falla (el nodo maestro)



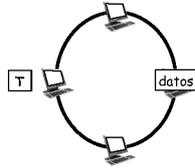
¿Cómo tomar turnos?

- Otra posibilidad es ir pasando una ficha (token) que denote quién puede hacer uso de canal:

- La ficha es pasada secuencialmente de un nodo al siguiente
- La ficha es en esencia una trama distinguida

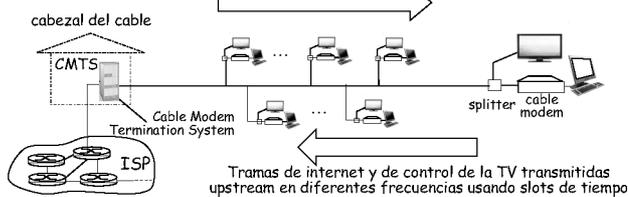
- **Análisis:**

- La ficha insume un costo extra
- La latencia se dispara
- Se introduce un punto único de falla (la ficha)



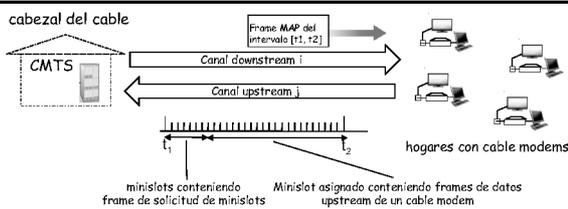
MAC en CATV

Tramas de internet, canales de TV y de control transmitida downstream a diferentes frecuencias



- **Downstream:** canales de **40 Mbps** (tipo broadcast)
- **Upstream:** canales de **30 Mbps**, donde todos los usuarios compiten por hace uso de los mismos

MAC en CATV



- **Data Over Cable Service Interface Spec.:**

- Se usa **FDM** en los canales tanto up como down
- Se usa **TDM** en upstream, donde los slots de datos están preasignados por el **CMTS** a través del frame **MAP**

MAC en CATV

- Los modems de cable solicitan aleatoriamente hacer uso del upload a través de los minislots
 - Nótese que más de una modem puede elegir usar el mismo minislot, produciendo una colisión
 - Como los modems son incapaces de detectar las colisiones, éstas se infieren ante la falta de respuesta por parte del **CMTS**
 - En caso de colisión, el modem hace uso de una técnica análoga a la espera exponencial de ethernet
- ¡¡¡Esta es la razón por la que anda tan mal!!!

Resumen de protocolos MAC

- Particionando el canal: por tiempo, por frecuencia, también usando códigos
- Por acceso aleatorio:
 - **ALOHA** puro, **ALOHA** con slots, **CSMA**, **CSMA/CD**
 - **CSMA/CD** se usa en ethernet
 - **CSMA/CA** se usa en **IEEE 802.11 (WiFi)**
- Tomando turnos:
 - Polling desde un nodo central, pasaje de tokens
 - Bluetooth, **FDDI**, token ring

¿Preguntas?