Temas 13 y 14

Acceso múltiple de paquetes: el protocolo Aloha

Eytan Modiano
Instituto Tecnológico de Massachusetts

Acceso Múltiple

- **Medio de transmisión compartido:**
  - un receptor puede oir a múltiples emisores
  - a un emisor lo pueden oir múltiples receptores

- **El problema principal del acceso múltiple está en establecer el canal entre los usuarios; los nodos no saben cuándo los otros nodos tienen datos que enviar:**
  - Es necesario coordinar las transmisiones

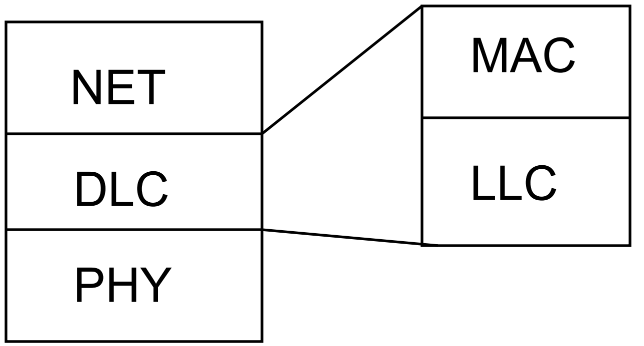
Ejemplos de canales de acceso múltiple

- **Redes de área local (LAN):**
  - **La tradicional *Ethernet***
  - **Tendencia reciente a que las LAN no sean de acceso múltiple**

- **Canales satélite**

- **Teléfono multiterminal**

- **Radio inalámbrica**



- **Control de acceso al medio (MAC)**
  - **Regula el acceso al canal**

- **Control del enlace lógico (LLC)**
  - **Todas las otras funciones DLC**

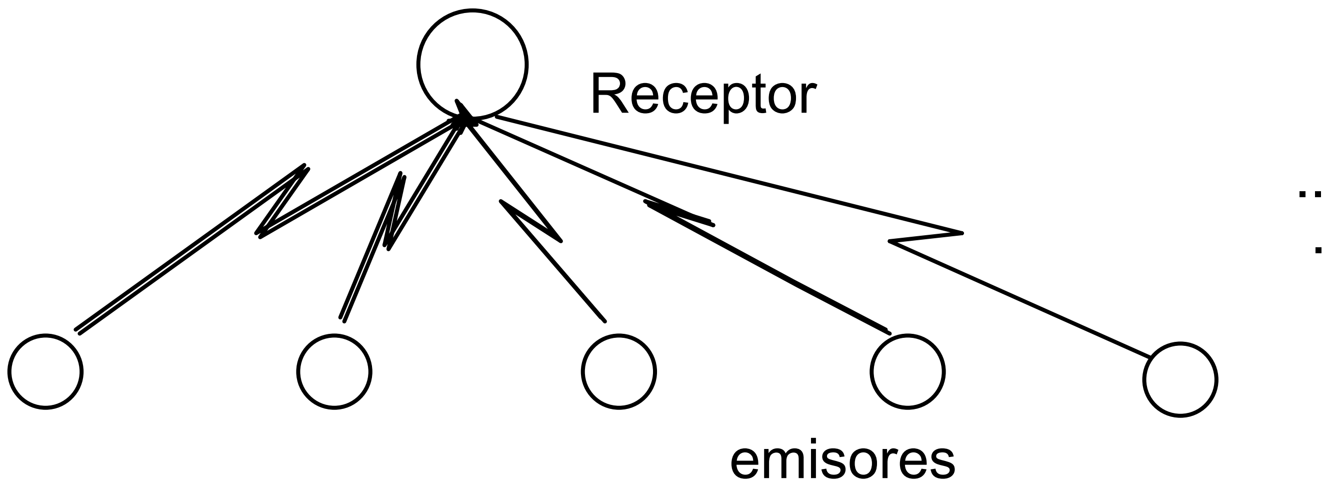
Enfoques del acceso múltiple

- **Asignación fija (TDMA, FDMA, CDMA):**
  - A cada nodo se le asigna una fracción determinada de ancho de banda
  - Equivalente a la conmutación de circuitos
  - Muy ineficaz para el tráfico con factor de bajo rendimiento

- **Sistemas de contienda:**
  - Sondeos
  - Reservas y gestión
  - Acceso aleatorio

Aloha

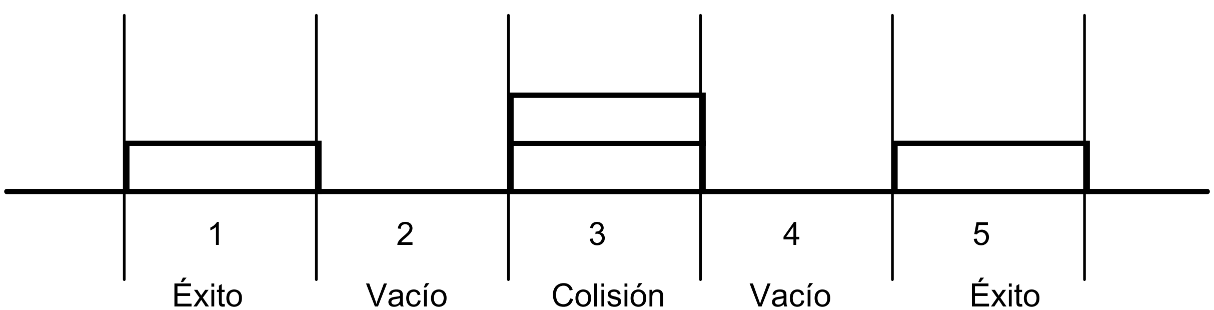
Un solo receptor; varios emisores



Ej.: Sistema de satelites, inalámbrico

Aloha con *slots* (divisiones o intervalos)

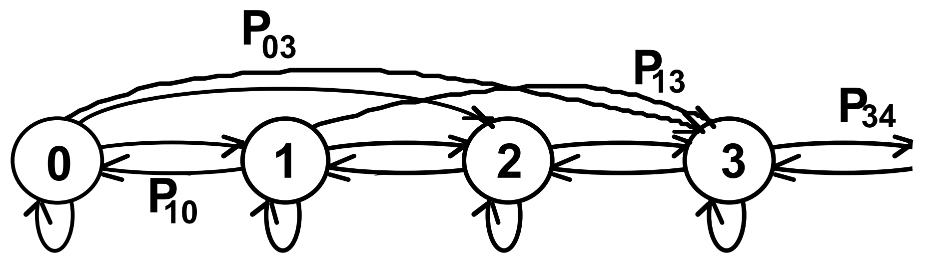
- **El tiempo se divide en "slots" de un paquete de duración:**
  - **Ej.: paquetes de tamaño fijo**
- **Cuando un nodo tiene un paquete para enviar, espera hasta el inicio del siguiente *slot* para enviarlo:**
  - **Requiere sincronización**
- **Si ningún otro nodo intenta transmitir durante ese *slot*, la transmisión tendrá éxito:**
  - **De lo contrario, se puede producir una "colisión"**
  - **Los paquetes que colisionan se retransmiten tras un tiempo de espera aleatorio**



Supuestos en el Aloha con *slots*

- **Llegadas externas de Poisson**
- **Sin captura:**
  - **Los paquetes que colisionan se pierden**
  - **También son posibles los modelos de captura**
- **Respuesta inmediata:**
  - **Vacío (0), éxito (1) y colisión (e)**
- **Si durante un *slot* llega un nuevo paquete, se transmite en el siguiente *slot***
- **Si se produce una colisión en una transmisión, el nodo se pone en modo de espera y va acumulando paquetes:**
  - **Mientras está en esta situación, transmite en cada *slot* con probabilidad q_r hasta alcanzar el éxito**
- **Los nodos son infinitos donde cada paquete entrante llega a un nuevo nodo**
  - **Equivalente a no hacer cola en un nodo (tamaño de la cola = 1)**
  - **Un supuesto pesimista tiene como resultado un límite inferior en el rendimiento de Aloha**

Cadena de Markov para el Aloha con *slots*



- El estado (n) del sistema es el número de nodos en espera:

  pi,i-1 = probabilidad de un intento en espera sin ninguna llegada nueva

  pi,i = probabilidad de una nueva llegada sin ningún intento en espera o ninguna llegada nueva sin ningún éxito

  pi,i+1= probabilidad de una nueva llegada con uno o más intentos en espera

  pi,i+j = probabilidad de J llegadas nuevas y uno o más intentos en espera o de J+1 llegadas nuevas y ningún intento en espera

- No existen probabilidades de estado estacionario:
  - La acumulación en espera tiende a infinito => sistema inestable
  - Más reciente

Aloha con *slots*

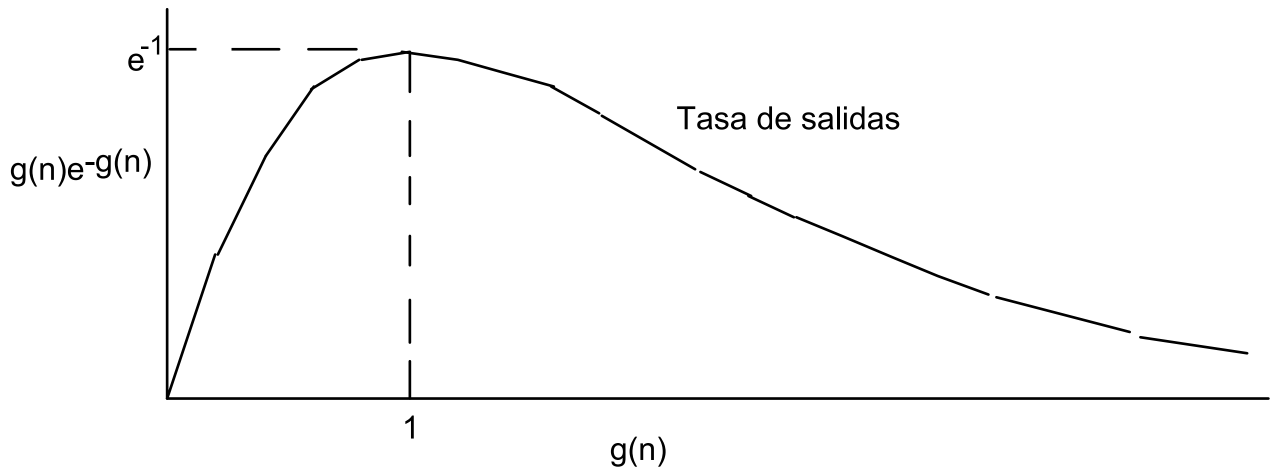
- **Sea g(n) la tasa de intentos (número esperado de paquetes transmitidos en un *slot*) en el estado n:**

$$g(n) = \lambda + nqr$$

- **El número de paquetes que se intentan enviar por *slot* en el estado n es aproximadamente una variable aleatoria de Poisson de media g(n):**
  - **P (m intentos) = $g(n)^m e^{-g(n)}/m!$**
  - **P (vacío) = probabilidad de que no haya ningún intento en un *slot* = $e^{-g(n)}$**
  - **P (éxito) = probabilidad de que haya un intento en un *slot* = $g(n)e^{-g(n)}$**
  - **P (colisión) = P (dos o más intentos) = 1 - P(vacío) - P(éxito)**

Tasa de transferencia de Aloha con *slots*

- **La tasa de transferencia es la fracción de *slots* que contienen transmisiones eficaces = P(éxito) = $g(n)e^{-g(n)}$**
  - **Cuando el sistema es estable, la tasa de transferencia debe ser igual a la tasa de llegadas externas (λ)**



  - **¿Qué valor de g(n) maximiza la tasa de transferencia?**

$$\frac{d}{dg(n)} g(n)e^{-g(n)} = e^{-g(n)} - g(n)e^{-g(n)} = 0$$

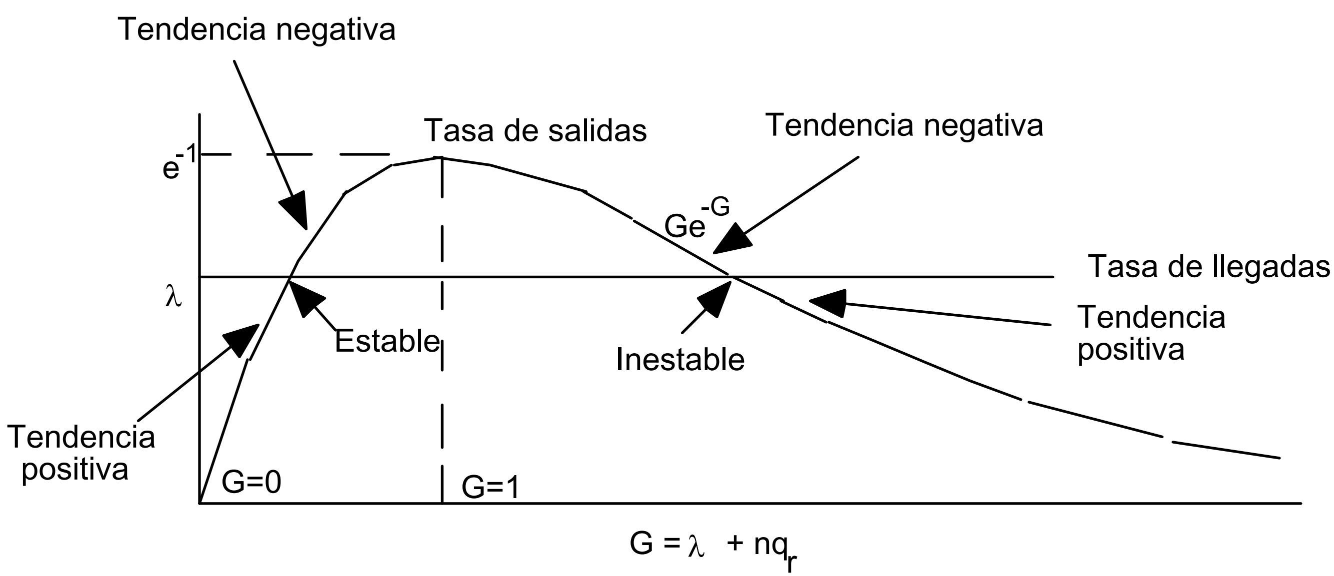
$$\Rightarrow g(n) = 1$$

$$\Rightarrow P(\acute{e}xito) = g(n)e^{-g(n)} = 1/e \approx 0.36$$

  - **g(n) < 1 => demasiados *slots* vacíos**
  - **g(n) > 1 => demasiadas colisiones**
  - **Si se puede mantener g(n) próximo a 1, se puede sostener una tasa de llegadas externas de 1/e paquetes por *slot***

Inestabilidad del Aloha con *slots*

- **Si la acumulación en espera aumenta hasta superar el punto de inestabilidad, entonces tiende a aumentar sin límite y la tasa de salidas cae hasta 0**
- **La tendencia en el estado n, D(n) es el cambio que se espera en el *backlog* durante un *slot* de tiempo:**
  - **$D(n) = \lambda$ - P(éxito) = $\lambda - g(n)e^{-g(n)}$**



Estabilización de Aloha con *slots*

- **Al elegir un valor pequeño de q_r aumenta la acumulación en espera a la que se produce la inestabilidad (dado que $g(n) = \lambda + nq_r$), pero aumenta también el retardo (dado que el tiempo medio para el reintento es 1/qr)**
- **Solución: calcular la acumulación en espera (n) a partir de las interacciones anteriores:**
  - **A partir de la acumulación en espera calculada, elegir q_r para mantener g(n) = 1**
    - **Suponer que todas las llegadas tienen que esperar y se acumulan inmediatamente**
    - **$g(n) = nq_r$, P(éxito) = $nq_r (1-q_r)^{n-1}$**
    - **Para maximizar P(éxito) elegir q_r = mín{1,1/n}**
  - **Cuando el cálculo de n es perfecto:**
    - **los vacíos se producen con probabilidad 1/e,**
    - **los éxitos con probabilidad 1/e; y**
    - **las colisiones con probabilidad 1-2/e.**
  - **Cuando el cálculo es demasiado grande, se producen demasiados *slots* vacíos**
  - **Cuando el cálculo es demasiado pequeño, se producen demasiadas colisiones**
- **Los nodos pueden utilizar la información de respuesta (0,1,e) para realizar las estimaciones:**
  - **Una buena regla consiste en aumentar la estimación de n en cada colisión y disminuirlo en cada *slot* vacío o con éxito**
    - **Obsérvese que el aumento en una colisión debería ser $(e-2)^{-1}$ veces tan grande como la disminución en un *slot* vacío**

Aloha con *slots* estabilizado

- **Suponer que en todas las llegadas se tiene que esperar y que éstas se acumulan inmediatamente:**
  - $g(n) = nq_r$ = tasa de intentos
  - $p(\text{éxito}) = nq_r (1-q_r)^{n-1}$

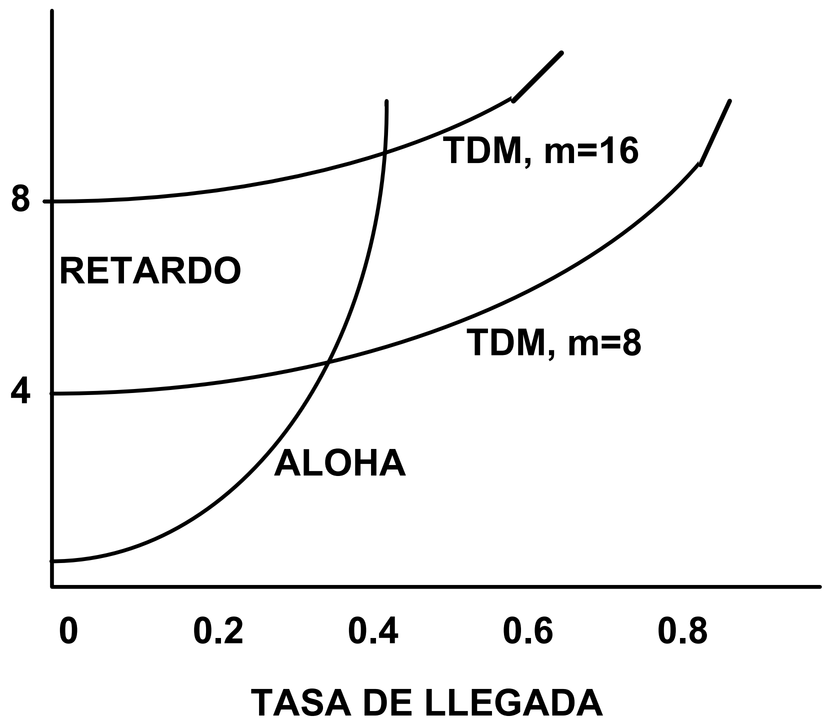
    Como tasa de transferencia establecer $g(n) = 1 \Rightarrow q_r = \text{mín}\{1, 1/n'\}$ donde n' es la estimación de n

  - **Sea n_k = la estimación de la acumulación en espera tras el *slot* k**

$$n_{k+1} = \begin{cases} \text{máx}\{\lambda, n_k+\lambda-1\} & \text{vacío o éxito} \\ n_k+\lambda+(e-2)^{-1} & \text{colisión} \end{cases}$$

  - **Se puede demostrar que es estable para $\lambda < 1/e$**

TDM frente a Aloha con *slots*

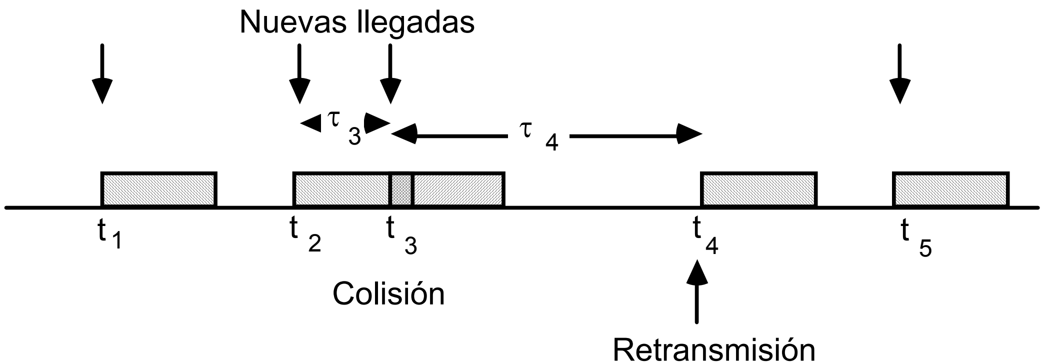


- **Aloha consigue retardos menores cuando las tasas de llegada son bajas**
- **Mientras Aloha es independiente del número de usuarios, TDM da lugar a retardos muy grandes cuando el número de usuarios es elevado**

Aloha puro (sin *slots*)

- **Las nuevas llegadas se transmiten inmediatamente (sin *slots*)**
  - **No es necesaria una sincronización**
  - **No es necesario que los paquetes tengan un tamaño fijo**

- **Un paquete en espera se intenta enviar otra vez tras un retardo aleatorio distribuido exponencialmente con alguna media de 1/x**
- **El proceso de llegada total es un proceso de Poisson de tiempo variable con tasa $g(n) = \lambda + nx$ (n = acumulación en espera, 1/x = tiempo medio entre las retransmisiones)**
- **Obsérvese que un intento sufre una colisión si el intento previo no ha terminado todavía ($t_i - t_{i-1} < 1$) o el siguiente empieza demasiado pronto ($t_{i+1} - t_i < 1$)**



Tasa de transferencia de Aloha

- **Un intento tiene éxito si los intervalos entre intentos en ambas partes son superiores a 1 (para paquetes de duración igual a la unidad):**
  - **P(éxito) = $e^{-g(n)} e^{-g(n)} = e^{-2g(n)}$**
  - **Tasa de transferencia (tasa de éxito) = g(n) $e^{-2g(n)}$**
  - **Para una tasa de transferencia máx. en g(n) = 1/2; tasa de transferencia = 1/2e ~ 0,18**
  - **Los problemas de estabilización son similares a los del Aloha con *slots***
  - **Las ventajas del Aloha sin *slots* son su simplicidad y la posibilidad de gestionar paquetes de distintos tamaños**

Algoritmos de separación (*Splitting*)

- **Un enfoque más eficaz para resolver las colisiones:**
  - **Interacción simple (0,1,e)**
  - **Idea principal: partir de que la colisión sólo afecta a dos paquetes:**

    Suponer que todos los otros nodos permanecen quietos hasta que se resuelve la colisión y que los nodos envueltos en ella transmiten con probabilidad 1/2 hasta que uno de ellos tiene éxito

    El otro nodo transmite en el siguiente *slot* posterior a este éxito

    El número esperado de *slots* para el primer éxito es 2, por lo que el número de *slots* que se espera que transmitan 2 paquetes es 3

    La tasa de transferencia durante los 3 *slots* = 2/3

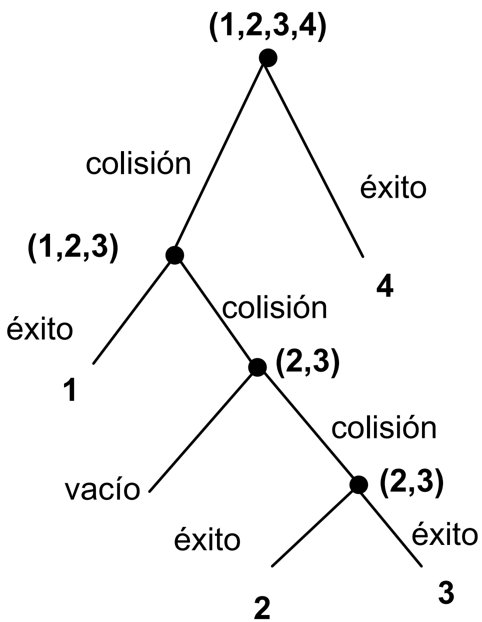
  - **En realidad, en la práctica, el algoritmo anterior no funciona:**

    No se puede suponer que participan en la colisión sólo dos usuarios

    Un algoritmo práctico debe contemplar la posibilidad de que se produzcan colisiones entre un número cualquiera de usuarios

Algoritmos en árbol

- **Tras una colisión, todas las llegadas nuevas y todos los paquetes almacenados que no participaron en la colisión esperan**
- **Cada uno de los paquetes que colisionan pasan a formar parte, aleatoriamente, de uno de los dos grupos (grupos Izquierda y Derecha):**
  - **Decisión a cara o cruz**
  - **El grupo Izquierda transmite durante el siguiente *slot,* mientras el grupo Derecha espera:**
    - **Si se produce una colisión, el grupo Izquierda se divide otra vez (se inicia el algoritmo)**
    - **El grupo Derecha espera hasta que se resuelve la colisión del grupo Izquierda**
  - **Cuando el grupo Izquierda está listo, el grupo Derecha transmite**



Obsérvese que tras el *slot* vacío, era seguro que se produciría una colisión entre (2,3) y, por tanto, se podía haber evitado

Hay muchas variaciones y mejoras del algoritmo original de división en árbol

Comparación de la tasa de transferencia

- Aloha puro estabilizado T = 0.184 = (1/(2e))
- Aloha con *slots* estabilizado T = 0.368 = (1/e)
- Algoritmo en árbol básico   T = 0.434
- Variación más conocida del algoritmo en árbol T = 0.4878
- Límite superior de cualquier algoritmo para resolver colisiones con interacción (0,1,e)   T <= 0.568
- TDM logra tasas de transferencia de hasta 1 paquete por *slot*, pero el retardo aumenta linealmente con el número de nodos