

---

# The Split and Merge Protocol for Interactive Video-on-Demand

**Wanjiun Liao  
Victor O.K. Li**

**Presentado por: David Vaquero López**

# Índice

---

- Autores
- Introducción
- Protocolo SAM (Split and Merge)
- Variaciones del esquema básico
- Resultados de simulación
- Conclusiones

# Autores

---

## ■ **Wanjiun Liao**



- Profesora asociada de National Taiwan University
- Amplio curriculum en investigación
- Ganadora de varios premios de prestigio en el ámbito investigador.
- Editora de Computer Networks y de IEEE Transactions on Wireless Communications
- Áreas de investigación: wireless networks, optical networks, and broadband Internet.

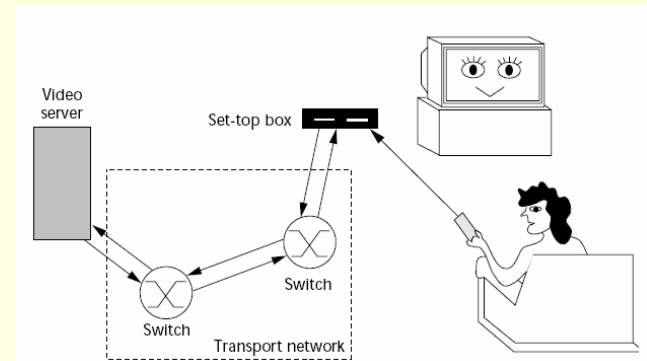
## ■ **Victor O.K. Li**



- Profesor catedrático de Ingeniería Electrónica en la Universidad de Hong Kong.
- Miembro de la IEEE desde 1992
- Más de 170 artículos de investigación.
- Áreas de investigación: high-speed communication networks, personal communication systems, distributed multimedia systems, distributed databases, queueing theory, graph theory, and applied probability.

# Introducción (I)

- VoD (Video-on-Demand) combina:
  - Calidad de servicio de TV cable
  - Capacidades de interacción de usuario (VCR)
- Un verdadero sistema VoD debe satisfacer:
  - Visión de video de alta calidad
  - Instante deseado
  - Interacciones de usuario (VCR)
- Competencia con los sistemas de video cotidianos
  - Se requiere un verdadero sistema VoD
  - Competencia en servicio y **precio**



# Introducción (II)

---

- Existen protocolos que intentan reducir coste por usuario
  - Basados en *batching*: comienzan el flujo al acumular suficientes usuarios o llegar a un límite de tiempo de espera.
  - No dan servicio de interacción
  - No son verdaderos sistemas VoD
- Protocolo SAM
  - Verdadero VoD
  - Mejora:
    - Recursos del sistema
    - Numero de usuarios simultáneos
  - Trabaja en distintas infraestructuras de red
  - Protocolo de red debe soportar **Multicast**
  - Sincronización de usuarios con “real stream”
    - Buffer de sincronización: situado normalmente en nodo de acceso

# Protocolo SAM (Split and Merge)

---

- Funcionamiento general
- Principios fundamentales
- Buffer de sincronización
- Flujo de operaciones
- Operaciones interacción de usuario
  - Jump-forward y jump-backward
  - Fast-forward y rewind
  - Pause y resume

# Protocolo SAM (Split and Merge)

## Funcionamiento General

---

- Sus principales objetivos son:
  - Proveer un verdadero servicio de VoD
  - Disminuir el coste por usuario
  - Aumentar el número de usuarios servidos con los mismos recursos de sistema y de red.
- SAM:
  - Operaciones split (salto a otro stream) y merge (fusionarse con un stream existente)
  - Habilitan cualquier operación de interacción de usuario
  - Protocolo:
    - Usuario visualiza video en un stream
    - Inicio interacción: virtual stream dedicado (split)
    - Fin interacción: stream real más cercano (merge)

# Protocolo SAM (Split and Merge)

## Principios Fundamentales (I)

---

- Interacciones usuario llevan menos tiempo que reproducción normal
- 2 tipos de video stream
  - Service stream (S stream): sirve a los usuarios reproducción normal
  - Interaction stream (I stream): satisface a los usuarios las interacciones VCR.  
1 I stream → 1 usuario



# Protocolo SAM (Split and Merge)

## Principios Fundamentales (II)

---

- Principios:
  - Proporciona verdadero servicio VoD
    - N usuarios por S stream
    - Petición de interacción por usuario
      - Split desde el S stream al I stream
    - Fin interacción
      - Merge al S stream más cercano
    - Pause no requiere I stream
  - Bloqueo de usuarios
    - Si todos los S stream están ocupados
    - Si admite petición de usuario el sistema
      - No bloqueará más el S stream para el usuario
      - Petición espera hasta que el recurso esté disponible
    - Mientras haya bloqueo y utilice S stream
      - Reproducción normal
      - Conmuta con el I Stream cuando recurso esté disponible

# Protocolo SAM (Split and Merge)

## Principios Fundamentales (III)

---

- Principios:

- Protocolo adaptable

- Aumenta demanda de 1 video → aumenta streams para ese video
    - Y viceversa
    - Gran eficiencia de SAM con videos de mucha demanda.

# Protocolo SAM (Split and Merge)

## Buffer de Sincronización (I)

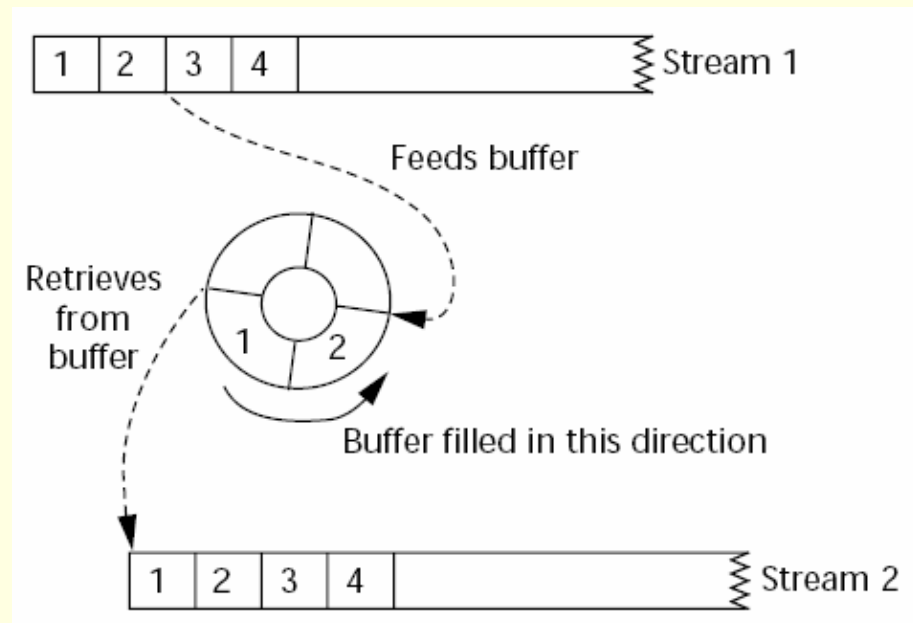
---

- Compartido por todos los usuarios
- Muy importante componente de SAM
- Buffer circular
- Se definen 2 operaciones:
  - Almacena segmentos de video
  - Recupera segmentos de video
- Se utiliza en la interacción de usuario:
  - Protocolo identifica el S stream más cercano a la petición del usuario
  - El S stream alimenta el buffer de sincronización que será recuperado por el nuevo virtual stream
  - Virtual stream: stream que es alimentado por el buffer de sincronización y utiliza el buffer como elemento intermedio entre el S stream y el usuario debido al offset existente.

# Protocolo SAM (Split and Merge)

## Buffer de Sincronización (II)

- S1: real stream
- S2: virtual stream
- División del video en segmentos etiquetados
- Utilizamos segmentos de S1 en vez de crear un nuevo S stream
- S1 almacena en buffer
- S2 recupera segmentos



# Protocolo SAM (Split and Merge)

## Buffer de Sincronización (III)

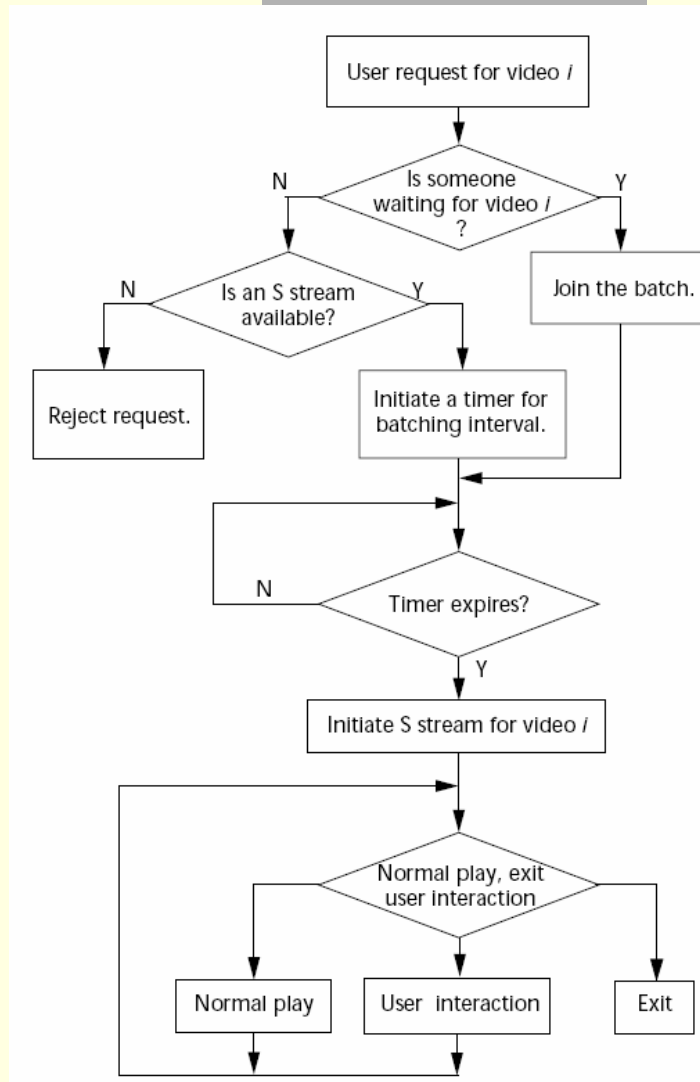
---

- Localización de buffer de sincronización
  - Redes grandes (residencial): nodos de acceso de usuarios
  - Redes pequeñas (LAN): servidor de video
- Tamaño de buffer
  - Para cada usuario
    - Tamaño =  $SB \times R_p$
    - SB: máxima duración del video (parámetro definido por el sistema)
    - $R_p$ : rate de codificación de video

# Protocolo SAM (Split and Merge)

## Flujo de Operaciones

- W: max tiempo de espera de un usuario para ver el video
  - Se diseña por parámetro
- Inicio batch
  - Bajo demanda
  - No periódicamente
- Variaciones de este esquema
  - Evitan retardo batching
- Operaciones Interacción
  - Pause, resume, fast-forward, rewind, jump-forward, jump-backward



# Protocolo SAM (Split and Merge)

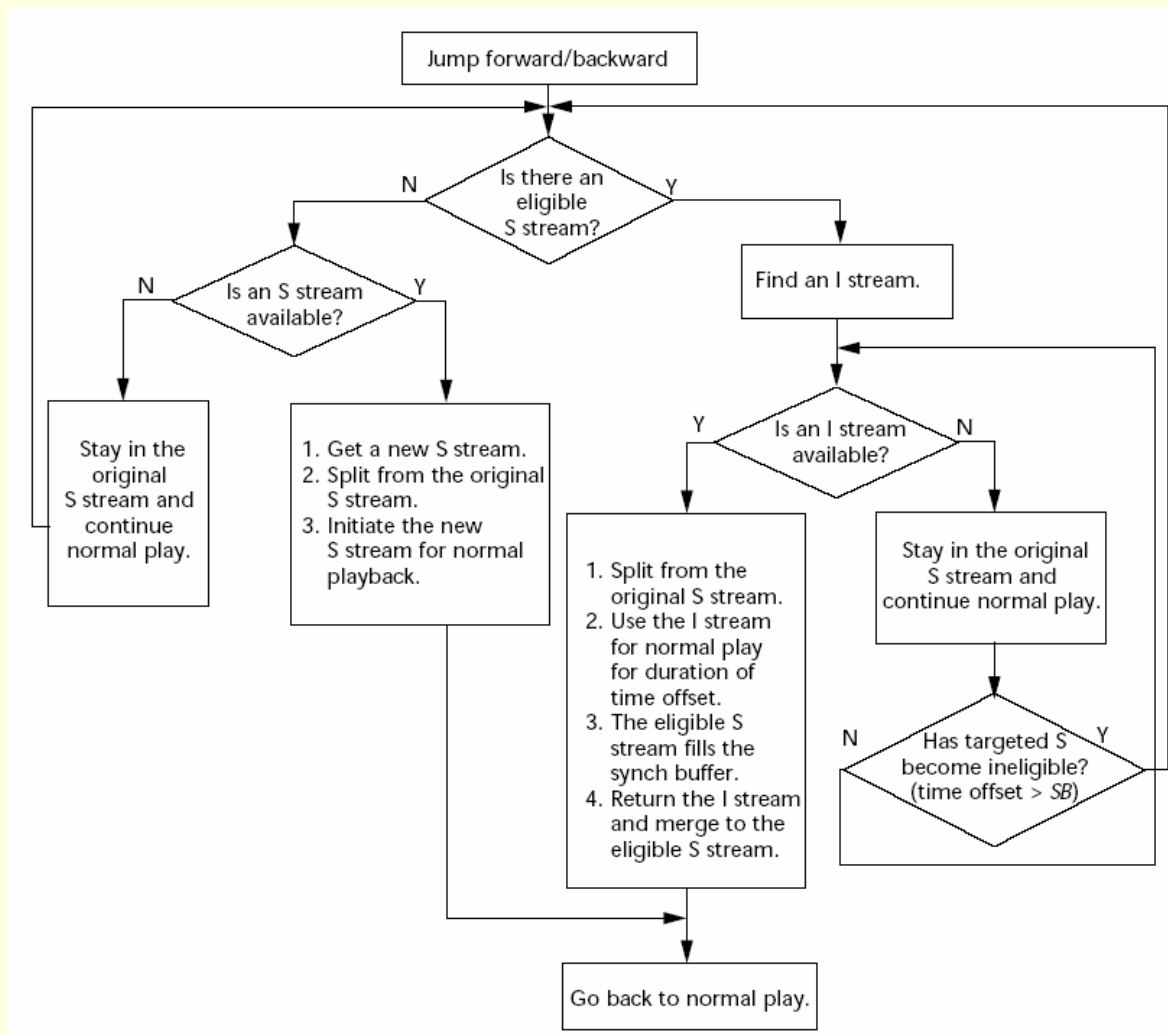
## Jump-forward y Jump-backward (I)

---

- Enfoque explicación en jump-forward
- **Elegible S stream:** S stream cuyo segmento apuntado por el 'play point' se ha reproducido antes de  $t_0$ , pero no más tarde que SB.
  - Caso de haber m elegibles: se elige el que tenga menor offset
$$t_{os} = t_0 - t_2$$
    - $t_0$ : instante en el que el usuario hace jump-forward
    - $t_1$ : segmento destino del usuario
    - $t_2$ : 'play point' en el elegible S stream

# Protocolo SAM (Split and Merge)

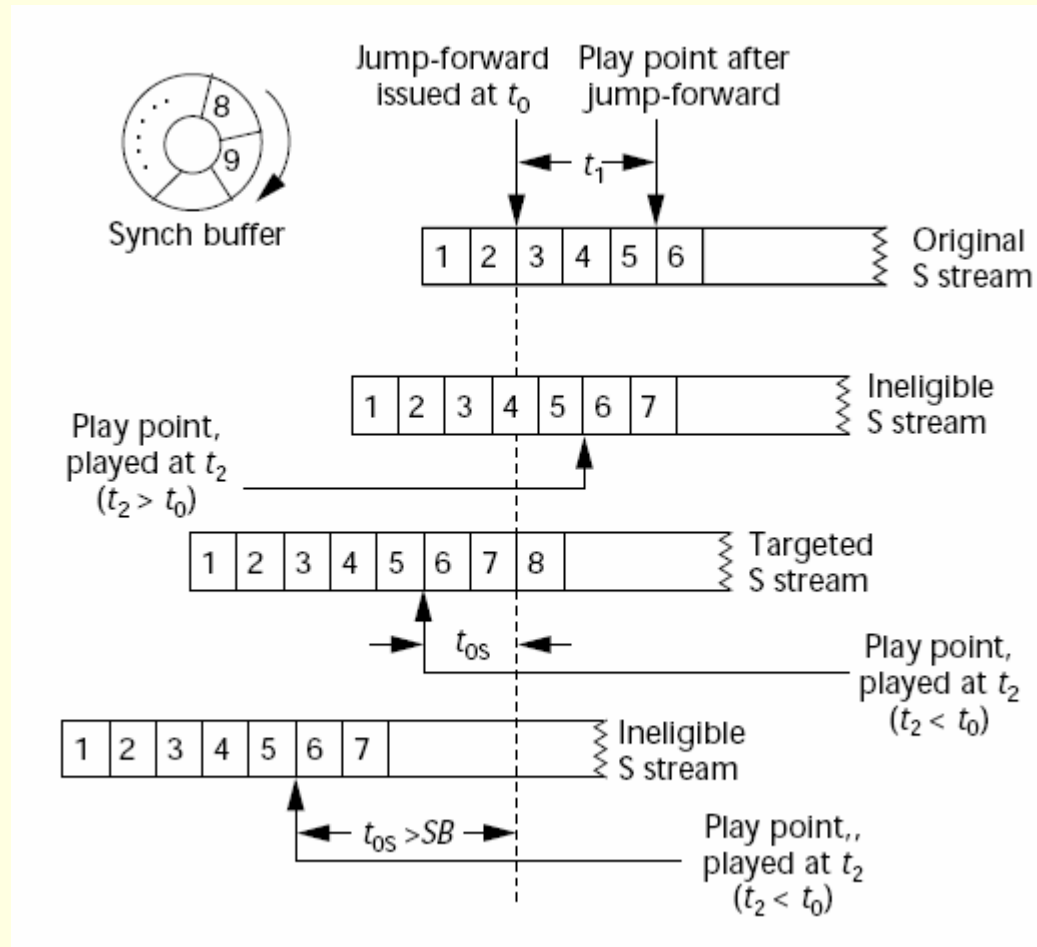
## Jump-forward y Jump-backward (II)





# Protocolo SAM (Split and Merge)

## Jump-forward y Jump-backward (III)



# Protocolo SAM (Split and Merge)

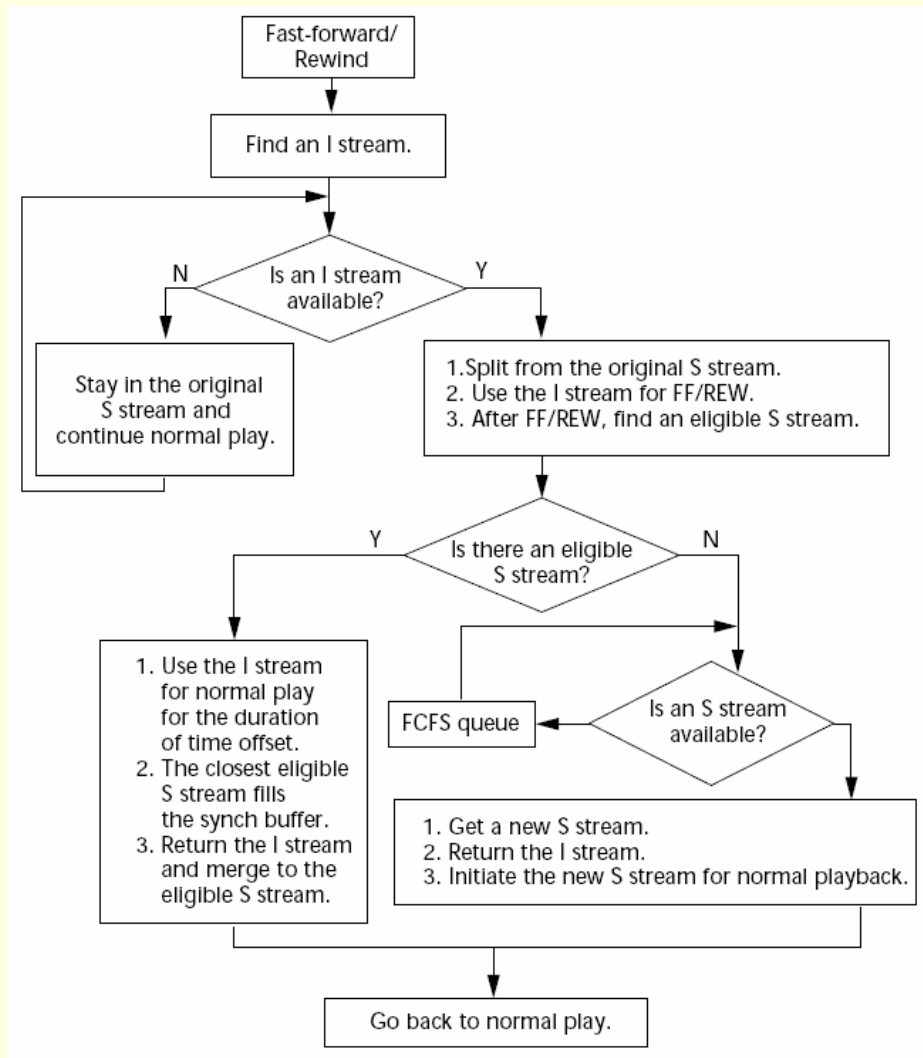
## Fast-forward y Rewind (I)

---

- Enfoque explicación en fast-forward
  - Caso de haber  $m$  elegibles: se elige el que tenga menor offset
  - $t_{os} < SB$ 
    - No overflow del buffer de sincronización
  - $t_2 < t_0 + d$ 
    - Tiene que haber sido reproducido en el elegible S stream
  - $t_0 + d$ : instante actual. Busco el elegible S stream
  - $t_2$ : 'play point' en el elegible S stream
  - $t_{os} = t_0 + d - t_2$

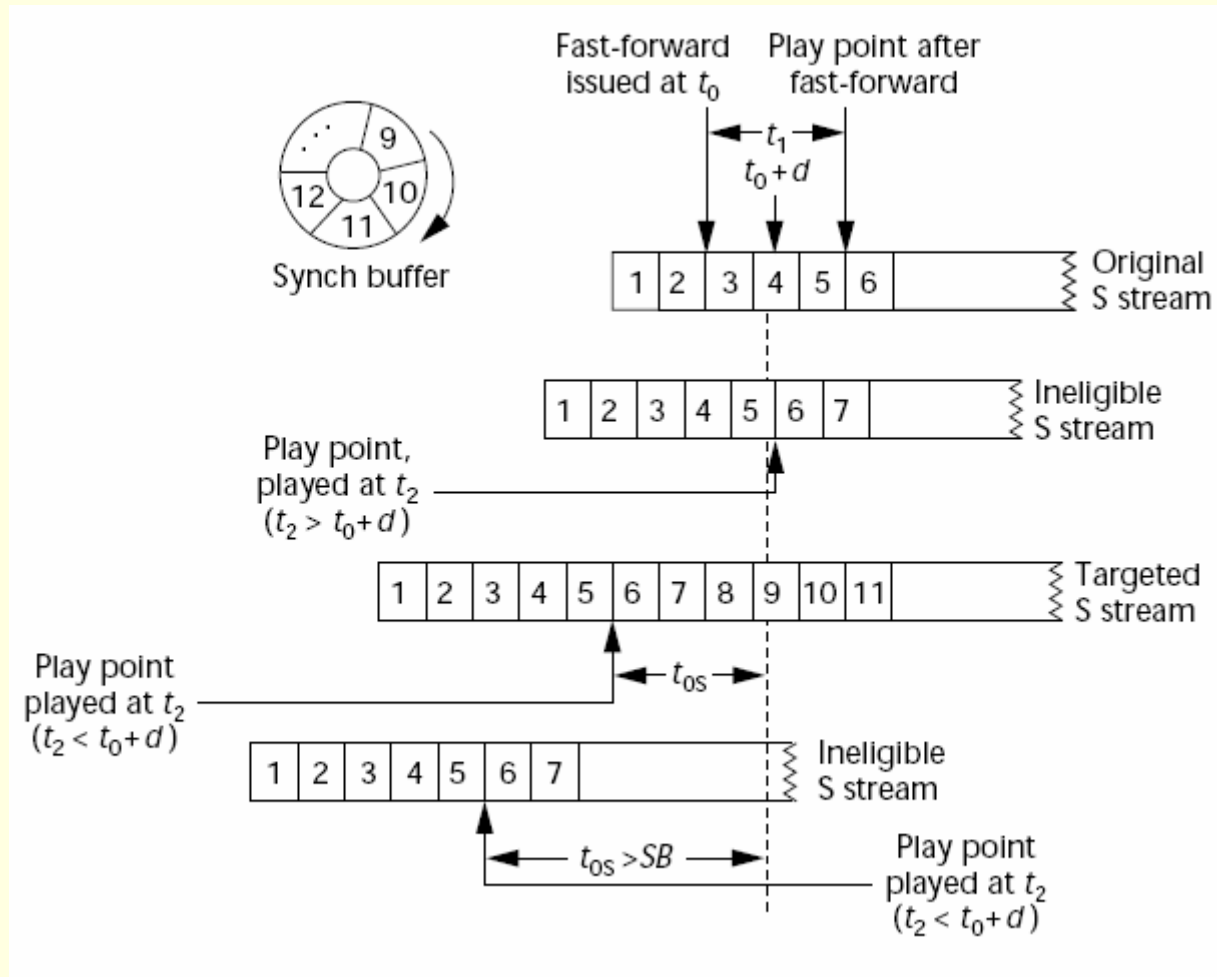
# Protocolo SAM (Split and Merge)

## Fast-forward y Rewind (II)



# Protocollo SAM (Split and Merge)

## Fast-forward y Rewind (III)



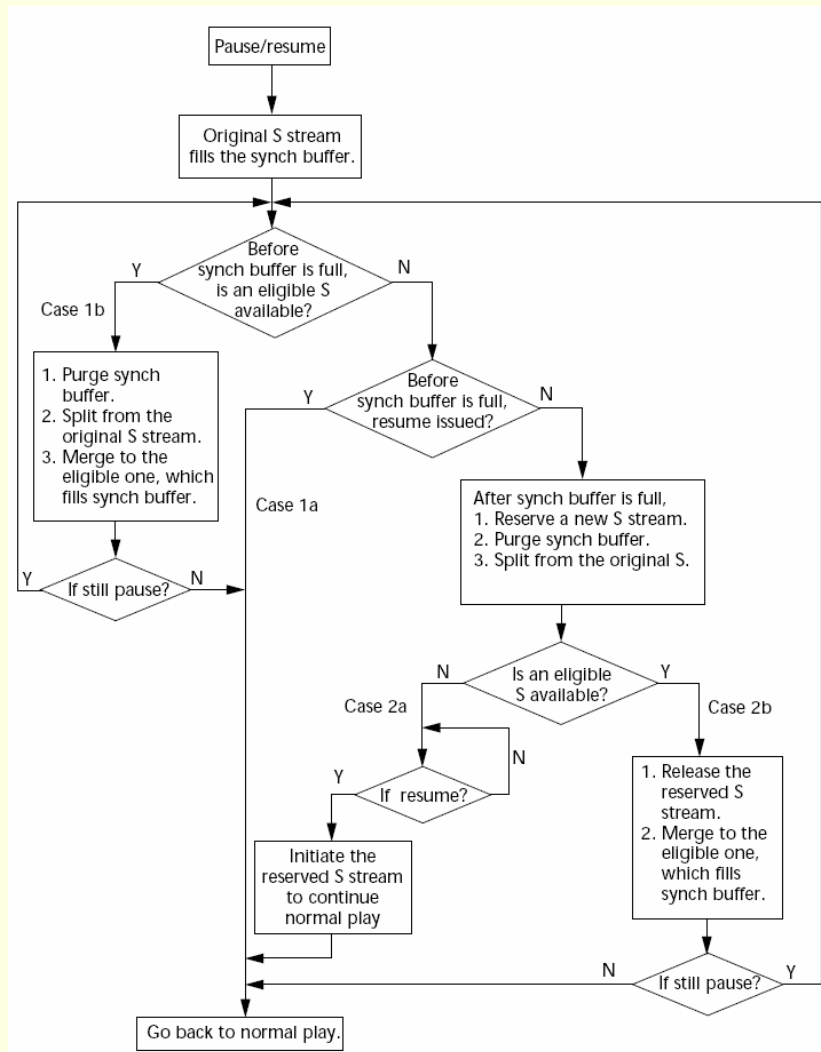
# Protocolo SAM (Split and Merge)

## Pause y Resume (I)

---

- No se requiere I stream
- Elegible S stream:
  - Tiene que haber comenzado la reproducción después del original stream
  - Tiene que haber reproducido el segmento apuntado por el 'play-point' antes de hacer resume
  - t mayor
  - d: periodo de pausa

# Protocolo SAM (Split and Merge) Pause y Resume (II)



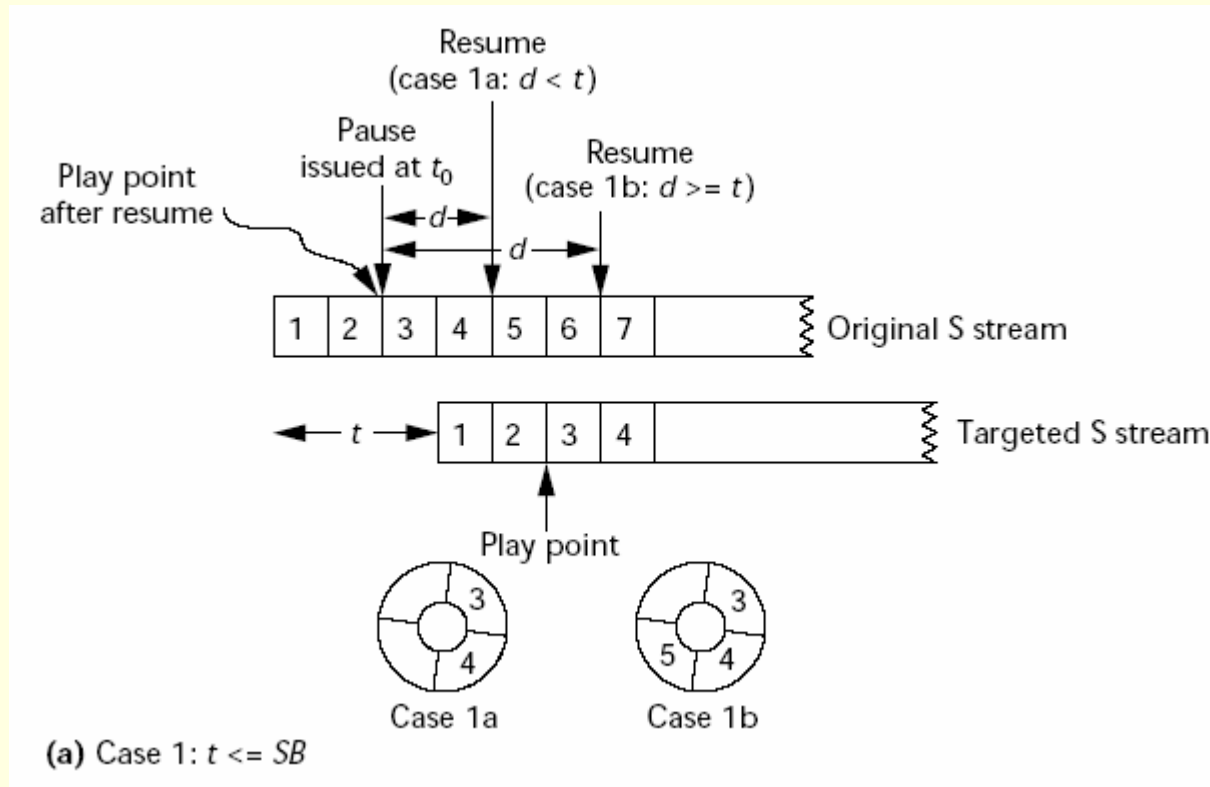
# Protocolo SAM (Split and Merge)

## Pause y Resume (III)

---

- Caso 1:  $t \leq SB$ 
  - a)  $d < t$  ;  $d < t \leq SB$ 
    - Almacena buffer se sincronización desde original stream
  - b)  $d \geq t$  ;  $t \leq SB$ 
    - El elegible S stream llega al 'play point' antes de hacer resume
    - $d' = d - t$ 
      - Si  $d' \leq SB \rightarrow$  caso 1b
        - Podemos almacenar los segmentos necesarios en el buffer de sincronización
      - Sino  $\rightarrow$  Caso 2
        - Hay que buscar un nuevo elegible S stream, ya que este no es válido (overflow en buffer)

# Protocolo SAM (Split and Merge) Pause y Resume (IV)





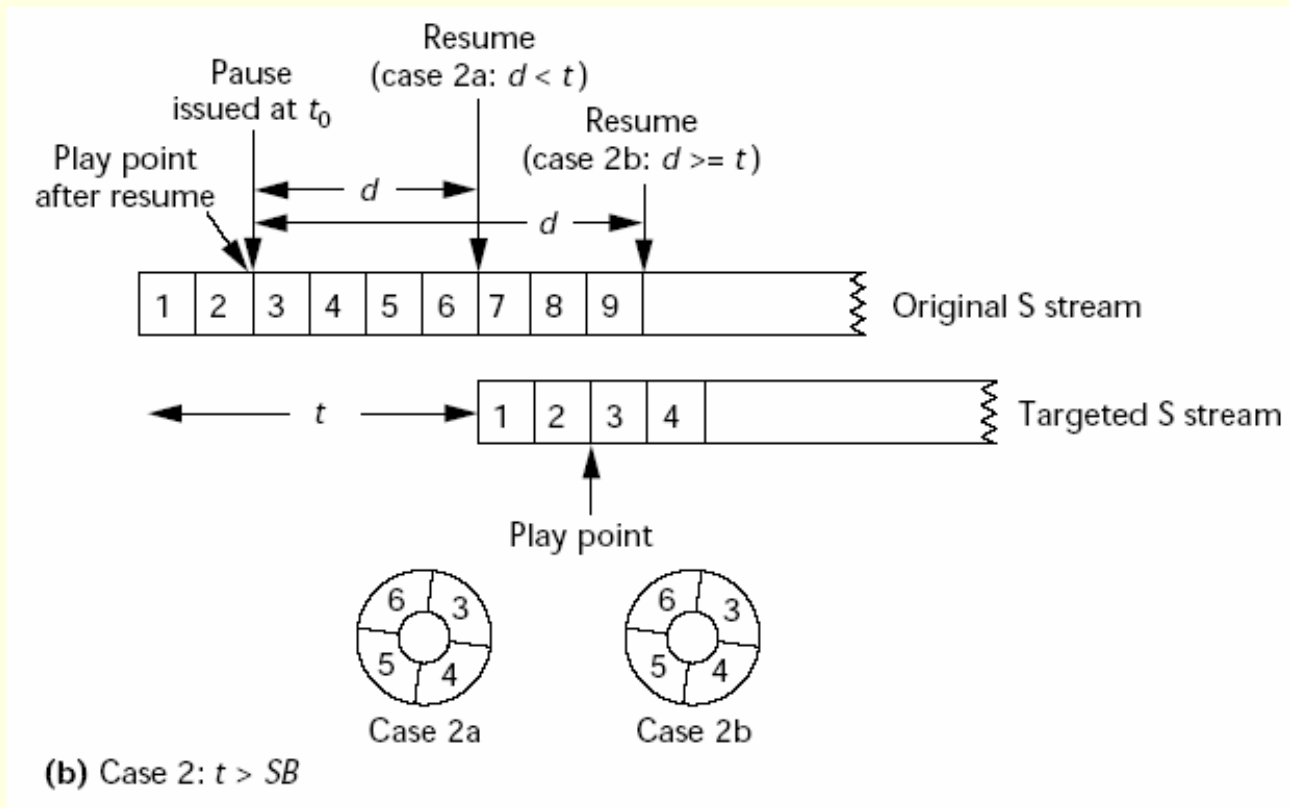
# Protocolo SAM (Split and Merge)

## Pause y Resume (V)

---

- Caso 2:  $t > SB$ 
  - a)  $d < t ; t > SB$ 
    - No es válido el elegible S stream porque no ha llegado al 'play point'
    - Coge un nuevo S stream porque la pausa ha terminado antes de que el elegible S stream haya llegado al 'play point'
  - b)  $d \geq t ; t > SB$ 
    - Libera nuevo S stream que comienza por el 'play point' y va almacenando segmentos en el buffer de sincronización
    - Así sucesivamente hasta que se ejecute el resume
  - Nota: al intentar utilizar un nuevo S stream, si hay un S stream que se pueda realizar un merge, se utilizará para optimizar recursos

# Protocolo SAM (Split and Merge) Pause y Resume (VI)



# Variaciones del Esquema Básico (I)

- No espera inicial de batching
  - Recibe petición
  - ¿Hay algún elegible S stream?
    - SI
      - Creación I stream durante  $t_{os}$
      - Merge al S stream que va alimentado el buffer de sincronización
    - NO
      - Crea un nuevo S stream
- Intervalos de batching ajustables
  - Aumenta demanda → Disminuye tiempo inicial batching
  - Cambio periódico intervalo de batching
    - Basándose en las peticiones ocurridas en el periodo anterior

# Variaciones del Esquema Básico (II)

---

- Un buffer de sincronización para múltiples virtual streams
  - Operaciones SAM → examinan S stream reales
  - Hay muchos virtual stream conectados al stream real a través del buffer de sincronización
  - Disminuye el número de buffer de sincronización → 1 buffer sirve varios virtual stream
  - Ej:
    - 1 buffer → offset 3 min
    - 2 virtual stream sobre el mismo real stream (1 min offset y 2 min. Offset respectivamente)
    - 1 entrada al buffer
    - N salidas a virtual stream con distintos offset

# Variaciones del Esquema Básico (III)

---

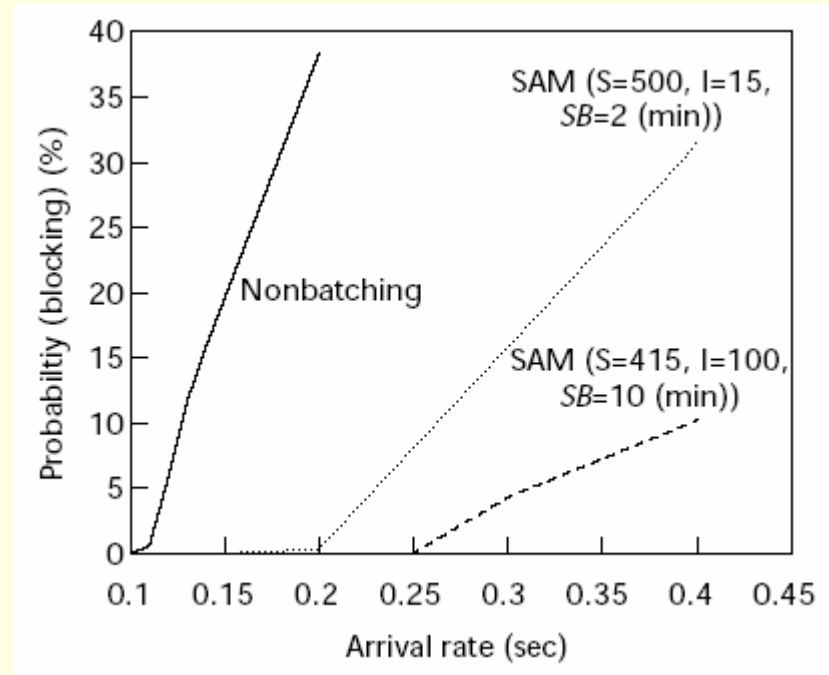
- Variante de la idea básica de batching
  - Inicio de contador de tiempo → primera petición
  - División de tiempo inicial en intervalos fijos de tiempo
  - Si durante un intervalo llega al menos una petición
    - Sistema inicia el S stream al final del intervalo
  - Disminuye tiempo de espera para videos no populares
  - Los videos populares no se aprecia diferencia
- Mecanismo de facturación según interactividad
  - Aumenta precio → Aumento calidad se servicio
  - Precios
    - Max: No tiempo inicial de batching y total interactividad
    - Medio: Tiempo inicial de batching y total interactividad
    - Min: Tiempo inicial de batching y limitada interactividad

# Resultados de Simulación (I)

- Sistema:
  - Programa en C corriendo en HP C 160
  - 24 horas de tiempo real
- Parámetros del sistema:
  - 30 videos disponibles
  - 120 min por video
  - Probabilidad progresiva de videos según popularidad (+ popular → + prob petición)
- Modelo de usuario
  - 30 min. Reproducción normal
  - A partir de los 30 min.
    - 75% interacción
    - 25% fin reproducción
  - Interacciones
    - Pause cada 5 min → distribución exponencial
    - Fast-forward o rewind → 0.5 min. Distribución uniforme entre 1 y 90 s.
    - Jump → 1 s. Distribución uniforme entre 1 y 1000 s.
  - Número total de streams: 515
  - Intervalo de batching: 10 min
  - Max buffer de sincronización: 1-10 min.

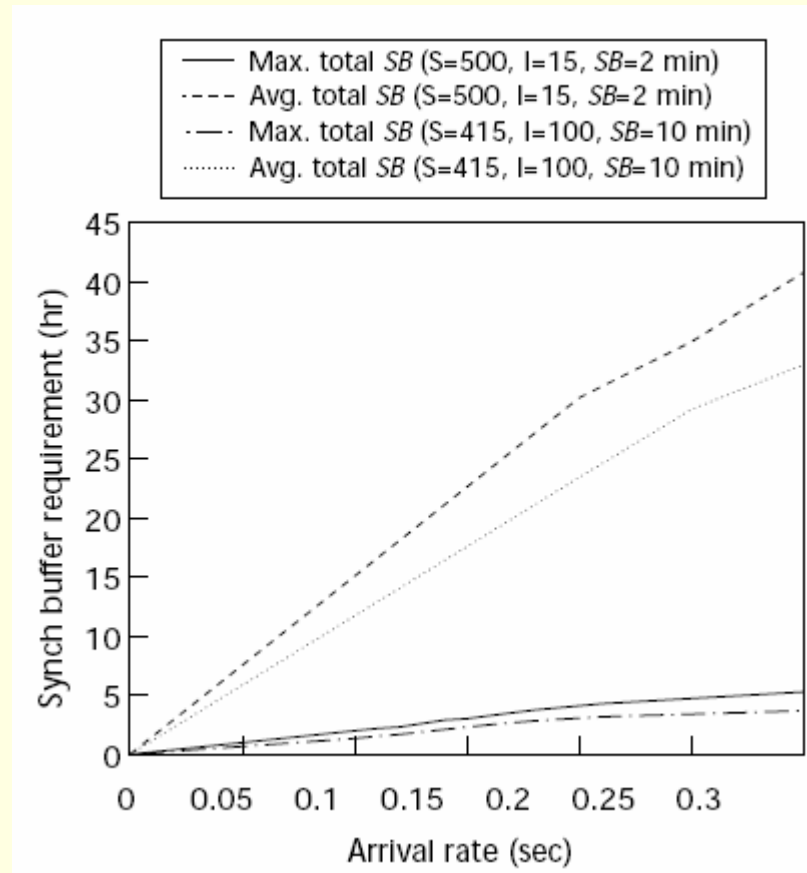
# Resultados de Simulación (II)

- Probabilidad de bloqueo frente al número de peticiones por segundo
  - Sistema nonbatching vs SAM
  - Nonbatching → más bloqueos (obvio)
  - Muestra la media para todos los videos en general
    - Caso de videos más populares → Aumenta bloqueos



# Resultados de Simulación (III)

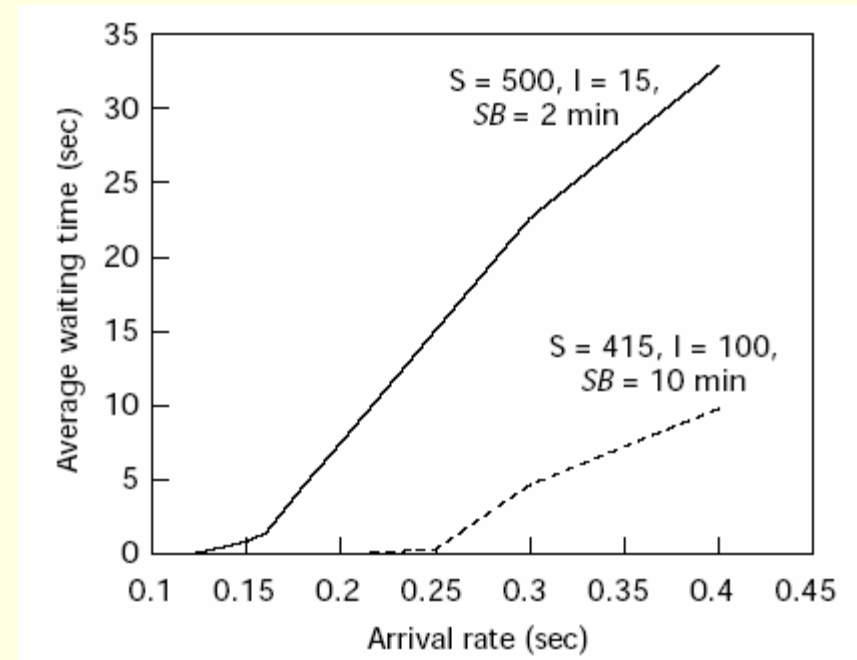
- Media total de buffer requerida frente al número de peticiones por segundo
  - Número pequeño por usuario
  - 8,3 segundos por usuario para rate 0.3
  - $0.3 \text{ pet/s} = 0.3 * 120 * 60 = 2160 \text{ peticiones}$ 
    - SB = 2min
    - Buffer / horas





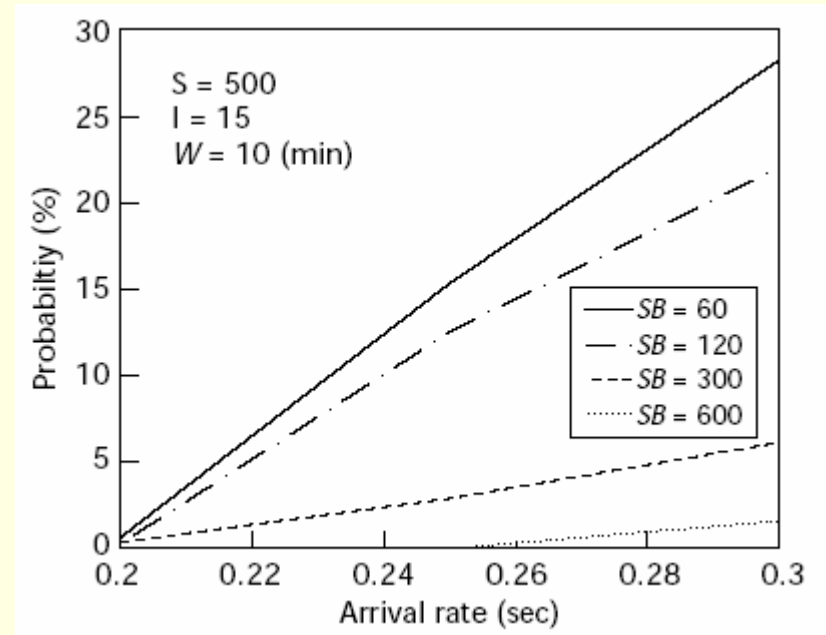
# Resultados de Simulación (IV)

- Media tiempo de espera de iteracción frente al número de peticiones por segundo
  - Aceptable excepto cuando hay alta carga

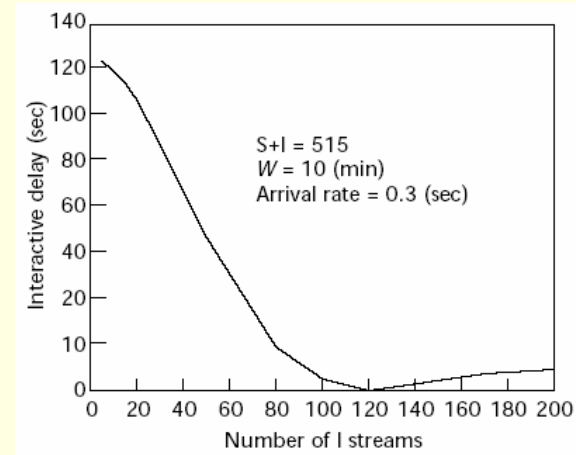
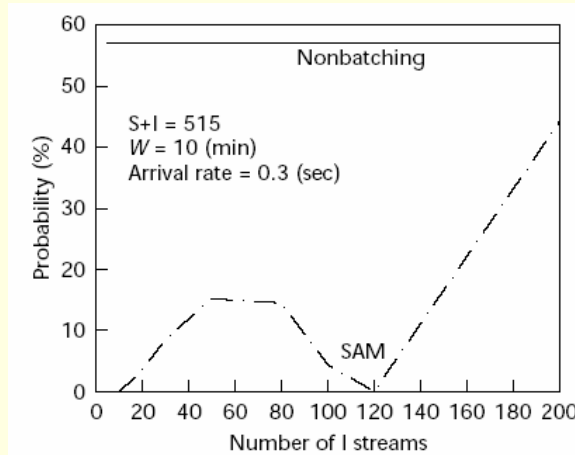
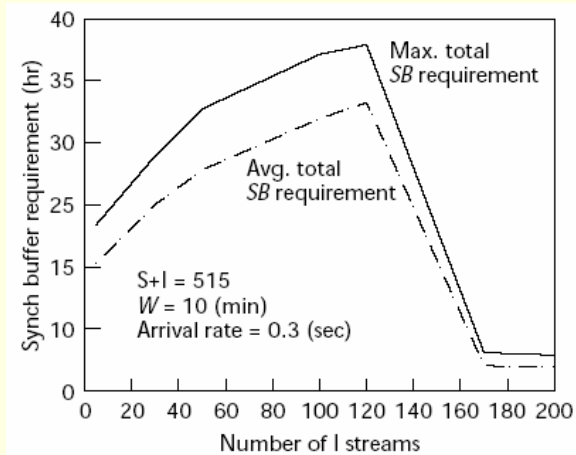


# Resultados de Simulación (V)

- Probabilidad de bloqueo frente al número de peticiones por segundo
  - Aumento tamaño de buffer → Disminuyen bloqueos



# Resultados de Simulación (VI)



- Probabilidad de bloqueo, tiempo espera interacción son mínimos con 120 I streams
- Tamaño de buffer de sincronización requiere su máximo.

# Conclusiones

---

- VoD → importante sistema de video multimedia distribuido en futuro
- Flexibilidad
  - Ver cualquier programa
  - En cualquier momento
  - Con interacciones de usuario VCR
  - Calidad TV por cable
- Comercial
  - Precio competitivo con video actual
  - SAM permite respecto a otros sistemas:
    - Múltiples usuarios a la vez
    - Aumento capacidad sistema
    - Disminución de costes por usuario
  - Precio dependerá de calidades exigidas por usuario

# The Split and Merge Protocol for Interactive Video-on-Demand

IEEE Multimedia, vol. 4, nº 4, Oct-Dec 1997

**Servicios en la Web y distribución  
de contenidos**

**UPNA**