

# Headlight Prefetching para Streaming Cooperativo en Entornos móviles.

Mauricio A. Ludueña G.  
Departamento de Electrónica  
Universidad Técnica Federico Santa María  
Valparaíso, Chile

**Abstract**— La masiva popularidad que poseen los servicios de media en línea como videos de música y eventos en vivo, junto con la proliferación de tecnologías 3G, han situado al Stream en dispositivos móviles como un tema de gran importancia para la investigación. Se deben tomar estrategias de manejo de datos para evadir desconexiones o interrupciones del servicio cuando los usuarios se mueven. En [1] se propone un framework de 2 niveles para aplicar nuevas técnicas para Streaming Cooperativo. Headlight Prefetching es expuesto como una técnica para la cooperación entre Streaming Access Points para lidiar con la incerteza del movimiento de la persona. Dynamic Chaining se usa para la cooperación entre usuarios móviles para maximizar la utilización de la caché. El siguiente informe muestra una simulación de la técnica de Headlight Prefetching, para disminuir el número de interrupciones y el tiempo total que tarda el usuario en descargar los archivos.

**Keywords**—Headlight; prefetching;SAP

## I. INTRODUCTION

El Streaming en dispositivos móviles es una tecnología prominente en el envío de información multimedia como deportes en vivo y videos de noticias. La llave para el suceso de este servicio es un buen manejo de la información a ser enviada. Como es común que gran cantidad de usuarios puedan pedir el mismo archivo multimedia a la misma hora (como noticias de última hora o la final del campeonato nacional), los modelos cliente-servidor pueden resultar en cuellos de botellas, poca utilización de la caché y retardos mayores. Broadcasting tampoco resuelve el problema, porque los usuarios pueden pedir la información en tiempos diferentes. Además, cuando un archivo multimedia comienza a reproducirse, los segmentos siguientes deben ser guardados para transmitirlos al usuario a tiempo y evitar interrupciones. Sin embargo, la movilidad de los usuarios y la imprevisibilidad de este movimiento complican aún más el problema de entrega de datos. Los segmentos guardados para un usuario pueden resultar inútiles si el usuario gira bruscamente para dirigirse a otra dirección. Por tanto, estrategias para el correcto y efectivo manejo de la información deben ser usados por todos los agentes que participan en el stream, incluyendo los usuarios, los streaming Access points (SAPs), y los servidores multimedia. Headlight Prefetching, descrito preliminarmente en [2] y profundizado en [1], está designado para la cooperación a nivel SAP, para lidiar con la incertidumbre del movimiento del cliente y el

requerimiento de servicio sin interrupción. Por cada cliente, se tiene una zona de guardado anticipado (prefetching) en forma de abanico en la dirección de movimiento del usuario. Todos los SAPs de la celda cubiertas por esta zona se coordinan para proveer el guardado anticipado para el usuario. La iluminancia virtual de la zona de prefetching en una celda determina el volumen de segmentos que deben ser guardados por el SAP de esa celda. El ángulo y la longitud pueden ser ajustados acorde a diferentes patrones de movimiento del usuario. Headlight Prefetching resuelve los problemas de identificar los SAPs responsables del prefetching, la cantidad de datos y la sincronización en un mecanismo simple y efectivo.

Para evaluación, se diseñó un primer acercamiento a este mecanismo usando un simulador Java, para comparar el tiempo requerido en la descarga y la cantidad de interrupciones del servicio tanto usando el método de Prefetching el tiempo sin este método.

El resto del informe se organiza de la siguiente forma: Sección 2 muestra la arquitectura de 2 niveles propuesta en [1] y entrega los desafíos del manejo de datos en esta arquitectura. Sección 3 introduce el concepto de Headlight prefetching, y las técnicas para el manejo de datos a nivel de SAP. Sección 4 se presenta la simulación en JAVA y presenta los resultados obtenidos por esta simulación. Sección 5 presenta las conclusiones del informe y el trabajo futuro.

## II. ARQUITECTURA DE SERVICIO DE STREAMING CON 2 NIVELES DE COOPERACIÓN

### A. Arquitectura

Se presenta la arquitectura de servicio ilustrada en la Fig. 1. El punto de origen de la información multimedia se denominan servidores de stream multimedia (streaming media servers, SMSs). Para el acceso inalámbrico, las células tienen su correspondiente SAP, conectado uno al otro a través de radio de alta velocidad. Cada SAP provee servicios multimedia inalámbricos para los usuarios móviles dentro de la celda.

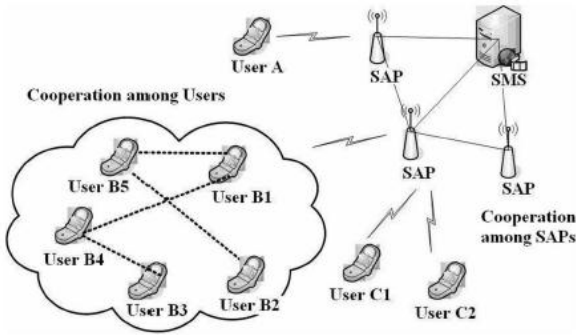


Fig. 1. Arquitectura de dos niveles de cooperación.

El primero de los niveles de cooperación es a través de los SAPs para proveer un manejo eficiente de los datos y disminución de las interrupciones. En general, sea  $P$  el costo promedio de unidad de stream entre usuarios,  $L$  el costo de acceso inalámbrico entre el móvil y el SAP, y  $M$  el acceso remoto desde el SMS. Entonces, basado en la naturaleza de la arquitectura y la carga del servicio, asumimos que  $P \leq L$  y  $P \leq M$ . En la mayoría de los casos,  $M < L$ , ya que el ancho de banda inalámbrico es generalmente menor al de enlace fijo. Si un segmento multimedia está disponible en un SAP, entonces nos toma un tiempo  $L$  en enviar el segmento. Si el segmento no se encuentra en el SAP, entonces un acceso remoto,  $M$ , debe agregarse además de  $L$ . Por esto, al reducir el número de accesos remotos es crítico ya que siempre es más caro que un acceso local.

Para efectos de esta investigación se considera un archivo multimedia como una secuencia de segmentos multimedia. Debido a la naturaleza del Stream, no es necesario obtener todos los segmentos a la vez. Una petición desde un dispositivo puede ser tomada como la petición de inicio de la secuencia de segmentos multimedia. Aquí se presentan varios desafíos para el manejo de datos:

- En General, un usuario puede pedir cualquier multimedia en cualquier lugar a cualquier hora. El SAP de la célula donde se encuentra el usuario debe encontrar y enviar el primer segmento al cliente tan pronto como sea posible para reducir el retardo.
- Una vez que el archivo se comienza a reproducir, el SAP debe guardar y transmitir los segmentos siguientes suficientemente rápido como para ir a la par con la velocidad de reproducción. Técnicas de prefetching y buffering son requeridas para evitar interrupciones.
- Un usuario puede moverse y cambiar de dirección en cualquier momento. Esta imprevisibilidad del movimiento solamente puede manejada por la cooperación con SAPs vecinos.
- Debido a la característica de reproducción continua, la mayoría de estos segmentos se ven una sola vez. Como la capacidad de almacenamiento es limitada, estos segmentos se descartan luego de ser visto. Se debiera considerar reusar estos segmentos en la ventana de tiempo.

### B. Beneficios.

El objetivo de esta arquitectura es proveer de técnicas efectivas para responder a estos desafíos de manejo de datos para el servicio de streaming. Diferentes niveles de cooperación proveen distintas oportunidades para el diseño de nuevas estrategias del manejo de datos. En particular, Headlight Prefetching está designado para resolver el problema de identificar los SAPs correspondientes y determinar los segmentos de prefetching en un framework simple. Específicamente, el esquema propuesto posee los siguientes aspectos:

- Cooperación a nivel SAPs mediante el headlight es simple, intuitiva y efectiva. Tanto la iluminación virtual como la zona de prefetching son directas y fáciles de computar.
- Headlight prefetching es flexible y ajustable. Se puede usar la forma y tamaño del abanico para controlar el rango de prefetching.
- Proveen adicionalmente los servicios de Headlight Shifting y Headlight Sharing para coordinación entre SAPs.

Como resumen, los servicios de streaming multimedia poseen desafíos significativos en el manejo de datos. El sistema de dos niveles de cooperación ayuda a la coordinación de los SAPs y al soporte mutuo de los usuarios que hicieron peticiones. Con esto, se puede obtener un nivel alto de calidad de servicio de stream.

### III. HEADLIGHT PREFETCHING

Una característica importante en el stream, como se comentó anteriormente, es la continuidad de reproducción. Una vez que esta empieza, los usuarios esperan fluidez en la reproducción. No es probable que obtener y transmitir en el momento se mantenga a la par con la velocidad de reproducción. Por tanto es necesario guardar de antemano los segmentos (a través de un búffer). Prefetching tradicional se hace manteniendo una ventana deslizante inmediatamente después del segmento actual. Esto es satisfactorio solo cuando el usuario se está moviendo en línea recta. Para patrones irregulares, el acercamiento tradicional puede fallar. El problema está en que si el usuario cambia de dirección y se mueve hacia un nuevo SAP, este último podría no tener nada preparado para el usuario. Esta incertidumbre del movimiento puede fácilmente resultar en una experiencia poco placentera al ver el video. Hacer que todos los SAPs cercanos hagan el prefetch puede conducir a un gran número de operaciones innecesarias. Aún con SAPs bien equipados, esto es en general poco costo-efectivo. Por tanto se necesita un mecanismo que identifique continuamente el set de prefetching para los SAPs. Aquellos SAPs a los cuales es más probable que se dirija el usuario debieran tener mayor probabilidad. Luego de identificar los SAPs, aún falta determinar la secuencia correcta de segmentos para guardar. Tener a todos los SAPs guardando el mismo set

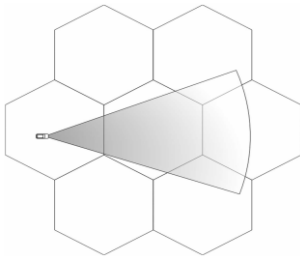


Fig. 2. Zona de headlight Prefetching

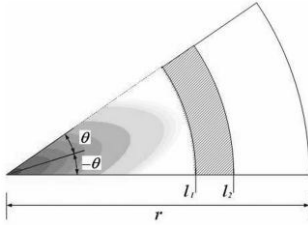


Fig. 3. Modelo de Headlight

de segmentos es poco satisfactorio e ineficiente. Nuevamente, debemos observar la incertidumbre del movimiento de los usuarios y el patrón determinando dinámicamente la asignación de segmentos. Los SAPs que estén más pronto a ser visitados debieran tener mayor número de segmentos a guardar. También se necesita una correcta sincronización para empezar o terminar el prefetching. De otra forma, los segmentos guardados llegarían muy temprano o muy tarde.

La idea de Headlight Prefetching es tener un mecanismo unificado para resolver los problemas descritos. Una zona de headlight prefetching (Fig. 2) es un área con forma de abanico en la dirección del movimiento del usuario, similares a las luces de un vehículo. Todos los SAPs que se traslapan con la esta zona son seleccionados para el prefetching. Esta zona está modelada por 2 parámetros (Fig.3). El radio  $r$  determina el largo de la zona, y el ángulo  $\theta$  controla el rango de cobertura. Ambos son dinámicamente ajustables. Por ejemplo, usuarios que se mueven más rápido necesitarán un mayor radio, y por otro lado, usuarios que tienden a deambular necesitan un mayor  $\theta$  para tener más SAPs listos. Notar que la zona de prefetching no es una conexión física, sino que es una zona puramente virtual. Sirve como una predicción de las posibles interacciones futuras entre los usuarios con las celdas.

Una vez que se identificaron los SAP, se necesita saber la asignación de segmentos. Para esto se ocupa la iluminancia. Siguiendo el mismo ejemplo que los autos, mientras más alejado esté el SAP en la zona de prefetching, menor es el grado de iluminancia. El área directamente en frente tiene la mayor luminosidad. Está característica calza perfectamente

con la asignación de segmentos. Como un usuario puede cambiar de dirección en cualquier momento, los segmentos guardados por los SAPs lejanos son menos probables que sean utilizados. Por tanto se le deben asignar menos segmentos para salvar costos. Los SAPs más cercanos al usuario son más

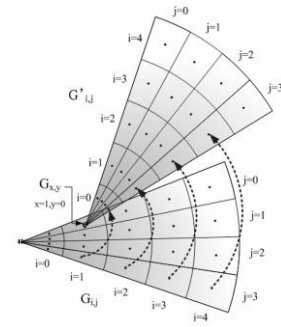


Fig. 4. Headlight Shifting, mapeo directo.

probables de tener la responsabilidad para proveer el servicio de stream. Por lo tanto deben alocar más segmentos para prevenir las interrupciones. Para esta asignación de segmentos, también se requiere considerar tanto la velocidad del móvil como la velocidad de reproducción.

### A. Headlight Shifting.

Un problema se produce cuando un usuario hace un giro brusco. En este caso gran parte del prefetching hecho en el SAPs anterior puede ser inútil, puesto que el usuario ya no se dirige en la dirección de este SAP. Para eso, [1] propone estrategias de giro, en las cuales se envían los segmentos de la SAP antigua a la nueva. Esto trae consigo 2 problemas:

- ¿Cómo mapeamos la fuente y el objetivo del giro?
- Diferentes SAPs pueden tener distinto número de segmentos guardados disponibles en el giro, ¿cómo se distribuye y balancea la carga del giro?

Se proponen entonces 3 estrategias:

- Mapeo Directo (Fig. 4), que es simplemente mapear la grilla de la nueva zona a la correspondiente grilla de la zona actual.
- Mapeo Traslapado, se ocupa para resolver el problema de que pasa si un usuario está por salir de una grilla. Este método consiste en el mapeo de grilla en formato one-to-many, es decir, una grilla en la antigua zona mapea varios puntos en la nueva zona.
- Estrategia de Mapeo por Distancia más Corta, que resuelve el problema del costo. Las estrategias anteriores son directas y fáciles de calcular, pero pueden no ser el costo de giro más barato. Esta nueva estrategia computa la distancia entre un SAP en la nueva zona y un SAP de la zona vieja, con lo que nos permite mapear el nuevo SAP al vecino de la zona vieja más cercano a este, que posea el segmento requerido.

### B. Headlight Sharing

Otra situación a considerar es el hecho de que multimedia de alta demanda es pedida no solo por 1 usuario, sino que por muchos. Esto puede llevar a que móviles con zonas de prefetching distintas tengan segmentos en común. Si las zonas

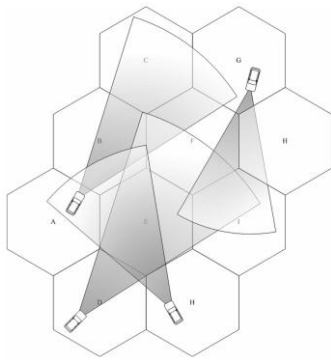


Fig. 5. Traslape de las zonas de Headlight.

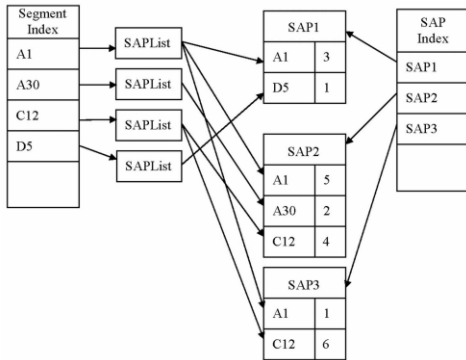


Fig. 6. Distribución de las tablas de index para Headlight Sharing.

se traslapan entre ellas (Fig. 5) es probable que podamos encontrar el segmento desde otra zona. A esta idea se le llama Headlight Sharing (compartir). Cuando un segmento es encontrado en un vecino, el costo de acceso remoto se puede omitir.

Para realizar esto, se propone una estructura índice en cada SAP (Fig. 6), que mantiene la información de los segmentos que contiene cada SAP. Cada segmento tiene un tiempo de expiración para indicar su período de validez. El SAP índice lleva un recuento de las tablas de índice disponibles, mientras que un índice de segmento mantiene un puntero a las entradas del índice del segmento en todas las tablas. Para evitar overhead, el mensaje índice se envía junto con el mensaje de prefetching.

#### IV. SIMULACIÓN

Se implementó un simulador en JAVA para comparar el tiempo promedio de descarga de segmentos con y sin headlight prefetching, usando los valores expuestos en la Tabla 1. Se simplificó la simulación a 1 móvil, para verificar en este el efecto del Headlight Prefetching. El programa tiene una serie de simplificaciones que se detallan a continuación:

- El móvil no gira (se mantiene siempre en la misma dirección), por tanto no se implementó Headlight Sharing.
- Como solo hay un móvil, no se implementó Headlight Sharing.

TABLA 1  
Parámetros de simulación

Parámetro	Valor
Número de móviles	1
Número de SAP	2
Radio de cobertura de SAPs	3[Km]
Tiempo entre SAP y SMS	0.05[s]
Tiempo entre SAP y móvil	0.05[s]
Tiempo de reproducción	1.5[s/seg]

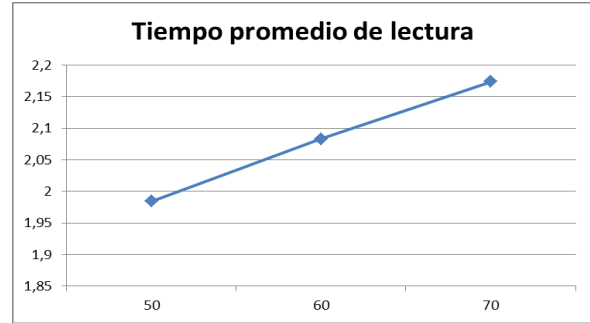


Fig. 7. Tiempo promedio de lectura de un paquete (en seg), en relación a la velocidad(en m/s).

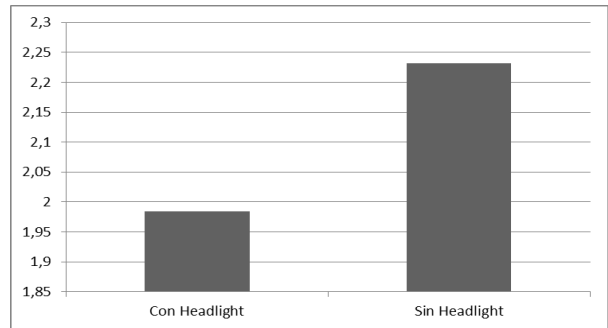


Fig. 8. Tiempo de lectura de un paquete, con y sin Headlight(en seg.), para una velocidad de 50[m/s]

- El número de SAPs es limitado, por lo que la simulación concluye cuando el móvil se encuentra fuera del área de cobertura de los SAP o terminó
- No se implementó el ángulo en la zona de prefetching, es decir, esta se comportó como una línea recta.

#### A. Resultados

Primero se simuló el tiempo promedio requerido para leer un segmento, dependiendo de la velocidad que se mueve el usuario. Podemos observar en la Fig. 7 que este tiempo tiene una dependencia lineal con la velocidad, ya que, si bien el tiempo de reproducción es el mismo, el tiempo requerido para la obtención de este paquete depende de la velocidad del móvil.

Luego se calculó el tiempo promedio necesario para la lectura de un paquete (recepción + reproducción). Podemos observar en la Fig. 8 que el tiempo promedio es menor usando Headlight Prefetching. Esto ocurre debido a que con Prefetching, el SAP 2 contiene ya los paquetes necesarios para entregar al móvil,

venía de la celda del SAP 1, y gracias a esto es capaz de impedir la interrupción del servicio. En el caso sin Prefetching, el SAP 2 comienza a descargar paquetes al momento de que detecta que el móvil está en su celda. Sin embargo, este retardo entre pedir el segmento al SMS y luego entregarlo al móvil es mayor a la velocidad de reproducción del archivo, por lo que aumenta el tiempo de lectura. Es debido a esto también que el ejemplo sin prefetching tiene una interrupción cuando el móvil cambia de celda.

#### V. CONCLUSIÓN Y TRABAJO FUTURO

Este trabajo logró afirmar que headlight prefetching ayuda a la disminución de lectura promedio, a la disminución de la interrupción, y en general, a una reproducción más expedita de la información multimedia.

A futuro, se propone implementar una mejor simulación, tomando en cuenta el ángulo del abanico. También se propone

aumentar la cantidad de móviles y SAP para ver la interacción entre ellos, además de implementar Headlight Sharing.. Adicionalmente, implementar Headlight Shifting, para poder continuar fluidamente con la reproducción aún si el móvil gira. Finalmente, simular Dynamic Chaining, no visto en este informe, para disminuir aún más el tiempo requerido para obtener el paquete.

#### REFERENCES

- [1] S.-Y. Wu, J. Hsu, and C.-M. Chen, "Headlight Prefetching and Dynamic Chaining for Cooperative Media Streaming in Mobile Environments," *IEEE Trans. On Mobile Computing*, Vol. 8, no. 2, pp. 173-87, 2009
- [2] S.-Y. Wu, J. Hsu, and C.-M. Chen, "Headlight Prefetching for Mobile Media Streaming," *Proc. Sixth Int'l ACM Workshop Data Eng. Wireless and Mobile Access (MobiDE)*, 2007.