



IRAC

Práctica 3

3.1 Integración de DASH en HTML5



Antonio Julián Alférez Zamora

Héctor Cuevas Esteban

Francisco García de la Corte

Índice

Índice de figuras.....	2
Resumen	3
1. Dynamic Adaptive Streaming over HTTP (DASH).....	4
1.1 Técnica DASH para streaming adaptativo	4
1.2 Media Presentation Description (MPD)	5
1.3 Algunas implementaciones	6
2. DASH-IF Reference player (dash.js)	7
3. Implementación	9
3.1 Infraestructura streaming	9
3.1.1 Telestream Wirecast	9
3.2 Cliente Web.....	10
4. Conclusiones	13
5. Bibliografía	14

Índice de figuras

<i>Ilustración 1.1: Esquema de segmentos enviados al cliente</i>	<i>4</i>
<i>Ilustración 1.2: Escenario genérico de implementación de DASH cliente-servidor</i>	<i>5</i>
<i>Ilustración 1.3: Escenario genérico de implementación de DASH cliente servidor</i>	<i>6</i>
<i>Ilustración 3.1: Pantalla principal Wirecast</i>	<i>10</i>
<i>Ilustración 3.2: Configuración de salida Wirecast</i>	<i>10</i>
<i>Ilustración 3.3: Cliente Web</i>	<i>11</i>

Resumen

En este trabajo se va a desarrollar y analizar el estándar Dynamic Adaptive Streaming over HTTP (DASH), así como una prueba de concepto de su implementación sobre HTML para un cliente Web.

Este estándar surge de la creciente demanda de contenido: clientes como teléfonos inteligentes, tabletas son normalmente usados con conexiones vía radio, es decir, cuyo ancho de banda es muy dinámico. El objetivo es realizar un estándar que permita reproducir contenido adaptándose al canal, aumentando así la experiencia de usuario.

En primer lugar, se realizará una introducción teórica para conocer su estado actual y así proveer de una base para la implementación práctica. Se tratan en una primera sección un contexto y explicación de la técnica DASH, junto con MPD (necesario para el cliente DASH) y algunas de las implementaciones existentes más relevantes.

A continuación, se entrará en detalle en el cliente desarrollado por el DASH Forum en JavaScript que posteriormente usaremos en la prueba de concepto.

Por último, se ha desplegado una pequeña infraestructura en la nube Microsoft Azure para realizar una transmisión streaming en vivo y poder reproducirla en un cliente web mediante DASH.

1. Dynamic Adaptive Streaming over HTTP (DASH)

1.1 Técnica DASH para streaming adaptativo

También conocido como MPEG-DASH (Moving Picture Expert Group – Dynamic Adaptive Streaming over HTTP) es una técnica estandarizada en 2011 para el streaming adaptativo (velocidad binaria adaptativa). También es la primera solución para streaming que se convirtió en un estándar. Usa la infraestructura existente de Internet para la entrega de contenido.

Se basa en un modelo de streaming controlado totalmente por el cliente, que solicita el contenido mediante servidores web HTTP convencionales, los cuales sirven segmentos que contienen pequeños intervalos del contenido en diferentes calidades, llamados segmentos. [1]

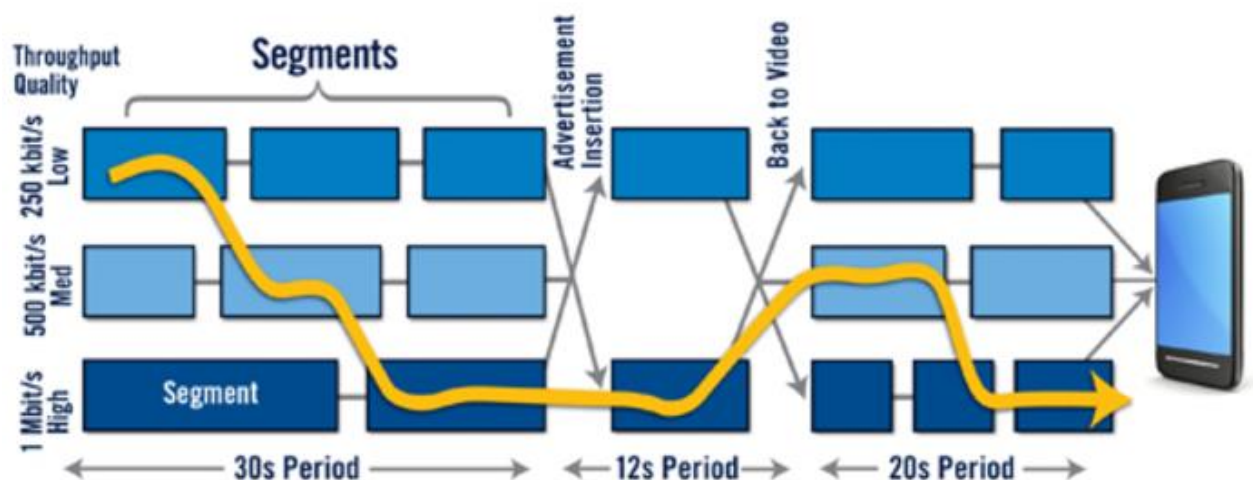
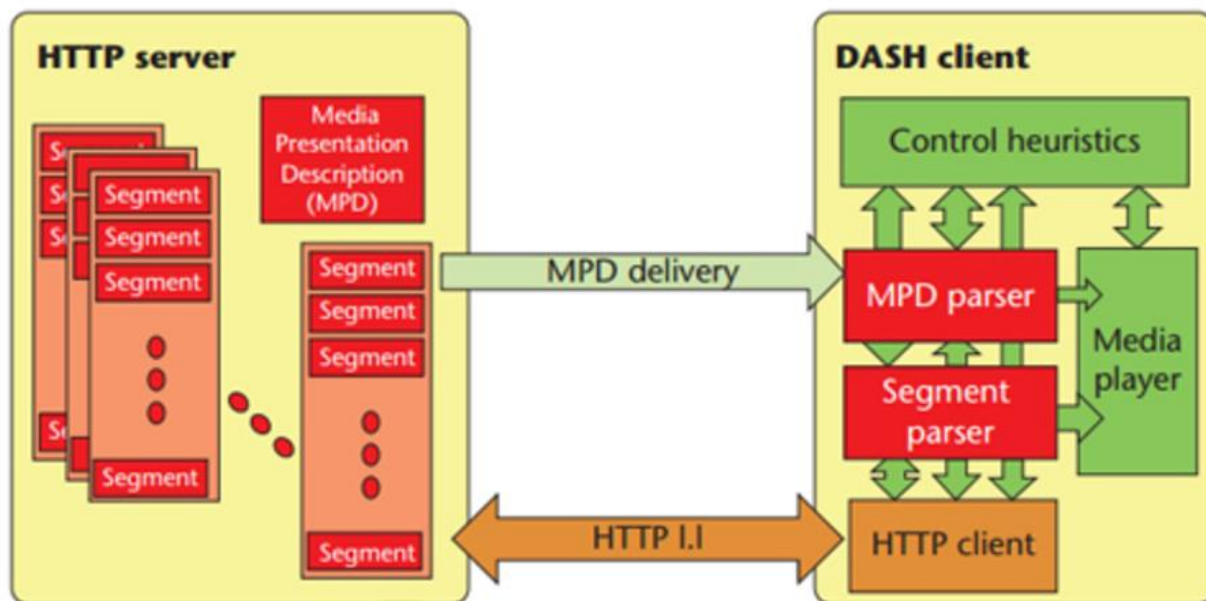


Ilustración 1.1: Esquema de segmentos enviados al cliente

Estos segmentos contienen varios segundos de vídeo, audio, subtítulos, etc. del contenido pedido por el cliente, que siguiendo la técnica DASH se piden los segmentos más adecuados a la capacidad del canal. Estas prestaciones están definidas en un fichero de metadatos llamado Media Presentation Description (MPD), fundamental para DASH. Siendo esto así, se define el siguiente escenario genérico:



Ilustraci n 1.2: Escenario gen rico de implementaci n de DASH cliente-servidor

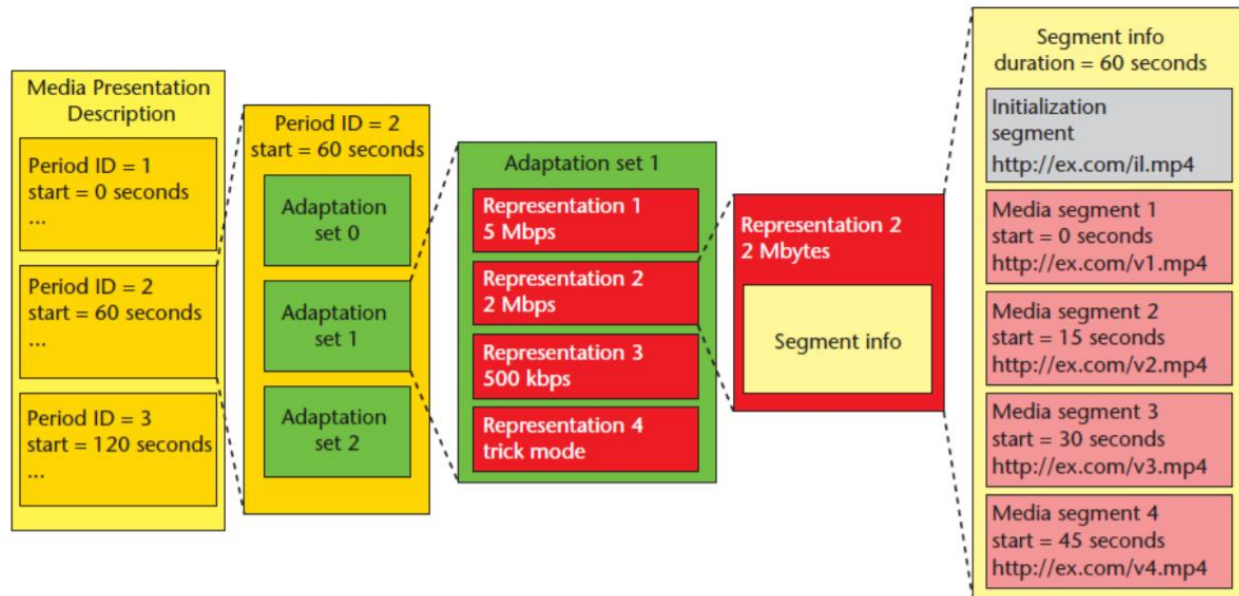
1.2 Media Presentation Description (MPD)

Se trata de un documento XML con los metadatos necesarios para que el cliente DASH pueda acceder al servicio de streaming al obtener los segmentos de este documento. Define los segmentos con sus calidades asociadas.

La estructura gen rica del MPD consiste en la jerarqu a: periods, adaptation sets, representations. Cada cual puede contener uno o m s del siguiente, formando la jerarqu a XML. De esta forma, el  ltimo nivel (representations) contiene la informaci n asociada a los segmentos, como es su localizaci n en tiempo y su URL.

Los periods representan un intervalo de tiempo en el contenido multimedia, y contiene varios adaptation sets.  stos, a su vez, contienen informaci n sobre uno o m s componentes multimedia y sus diferentes formas de codificaci n: representations. Las representations contienen las alternativas de codificaci n de los componentes multimedia, diferenci ndose en el bitrate que van a generar y los segments que el usuario DASH ha de pedir.

Por 3ltimo, estos segments son las partes en las que se ha dividido el stream. Contienen unas direcciones (URI) que referencian el contenido alojado en un servidor. Estas direcciones son accedidas por los usuarios mediante peticiones HTTP-GET. [2]



Ilustraci3n 1.3: Escenario gen3rico de implementaci3n de DASH cliente servidor

1.3 Algunas implementaciones

Siendo MPEG-DASH un proyecto de c3digo abierto, muchos terceros han podido implementar sus propias soluciones e implementarlas. De hecho, est3 disponible de forma nativa en Android mediante ExoPlayer, as3 como en muchas televisiones y Chromecast. Youtube y Netflix implementan MPEG-DASH.

En cuanto a los navegadores web, MPEG-DASH no es directamente compatible con HTML5 pero s3 a trav3s de implementaciones con JavaScript, incluso para v3deo 360 en directo.

Como clientes y librer3as, tenemos Dash.js, que es la implementaci3n oficial de Dash Industry Forums Reference, Shaka de Google (c3digo abierto) o Videogular para AngularJS.

2. DASH-IF Reference player (dash.js)

DASH Industry Forum (DASH-IF) es una organización sin ánimo de lucro dedicada a la adopción de MPEG-DASH, y creadora de dash.js. Es un conglomerado de compañías que trabajan juntas para la correcta implementación del protocolo.

Dash.js es una iniciativa del DASH-IF que establece un framework que usa MPEG-DASH usando librerías JavaScript en el lado del cliente. Tiene como objetivo la creación de una librería JavaScript de código abierto con las siguientes características: [3]

- Robustez para el mundo real
- Algoritmos de adaptación más eficientes
- Libre para el uso comercial
- Compatible con cualquier códec y navegador
- Buenas prácticas de MPEG-DASH
- Soporte de un gran abanico de funciones, como eventos, períodos múltiples o DRM entre navegadores distintos

Su documentación para el uso e integración en cualquier proyecto se puede encontrar en el siguiente enlace:

<http://cdn.dashjs.org/latest/jsdoc/index.html>

En el enlace se muestra el mismo contenido que aparece en el archivo readme del proyecto en GitHub. Aparece información relevante del proyecto:

“Un cliente de referencia para la implementación de MPEG-DASH vía JavaScript y navegadores implicados. Si su objetivo es usar el código del reproductor sin contribuir al proyecto, entonces use la rama MASTER, la cuál contiene los releases públicos, estables y aprobados.”

Siguiendo esto, y lo comentado en otro apartado llamado “Quick Start for Users” y otro llamado “Getting Started”, hemos llevado a cabo un proyecto propio especificado en el siguiente capítulo de implementación, además de la lógica de servidor propia.

La implementación básica para que funcione el “Reference Player” trata del siguiente código HTML y JavaScript, de acuerdo con lo especificado por el DASH-IF:

```
<!doctype html>
<html>
  <head>
    <title>Dash.js Rocks</title>
    <style>
      video {
        width: 640px;
        height: 360px;
      }
    </style>
  </head>
  <body>
    <div>
      <video id="videoPlayer" controls></video>
    </div>
    <script src="yourPathToDash/dash.all.min.js"></script>
    <script>
      (function(){
        var url = "http://dash.edgesuite.net/envivio/EnvivioDash3/manifest.mpd";
        var player = dashjs.MediaPlayer().create();
        player.initialize(document.querySelector("#videoPlayer"), url, true);
      })();
    </script>
  </body>
</html>
```

Las primeras líneas establecen una página web HTML básica, en la que hay un reproductor de vídeo con un tamaño especificado en el código y un identificador para el vídeo.

Las etiquetas <script> son las que definen el funcionamiento. Una primera etiqueta llama al código realizado por el DASH-IF: dash.js propiamente dicho. La segunda etiqueta obtiene una URL para acceder a un MPD (descrito en el anterior capítulo), que podría especificarse por el usuario en un formulario muy fácilmente, crea e inicializa el reproductor y lo asigna al campo de vídeo con el identificador usado previamente.

3. Implementación

En esta parte del trabajo, se ha desarrollado una pequeña plataforma de streaming en vivo que codifica en DASH y un cliente web que permita reproducir el contenido.

3.1 Infraestructura streaming

En primer lugar, hemos usado la nube Microsoft Azure para montar el servicio, más concretamente su solución Azure Media Services, una plataforma para la administración y entrega de contenido multimedia [4]. Esta solución nos permite enviar contenido a través del protocolo RTMP con velocidades múltiples y así poder distribuirlo aprovechando el estándar DASH, a través de un CDN también parte de la solución.

A grandes rasgos, esta plataforma nos provee de una dirección a la que enviar el flujo RTMP de la forma:

`rtmp://mychannel-irac08p3.channel.mediaservices.windows.net:1935/live/{identificador}`

que posteriormente configuraremos en el software codificador. Además, también nos dará una dirección al fichero Media Presentation Description (MPD) que introduciremos en el cliente para la reproducción:

`http://irac08p3.streaming.mediaservices.windows.net/{identificador}.ism/manifest(format=mpd-time-csf)`

Por otra parte, para ser exhaustivos con el objetivo del trabajo, también hemos configurado un CDN en Azure para una entrega de contenido más realista.

3.1.1 Telestream Wirecast

Se trata de un software de generación de contenido y distribución que hemos escogido porque es fácilmente configurable para Azure Media Services [5]. Permite, entre otras características, configurar transmisión en diferentes calidades y diferentes fuentes de vídeo y audio.

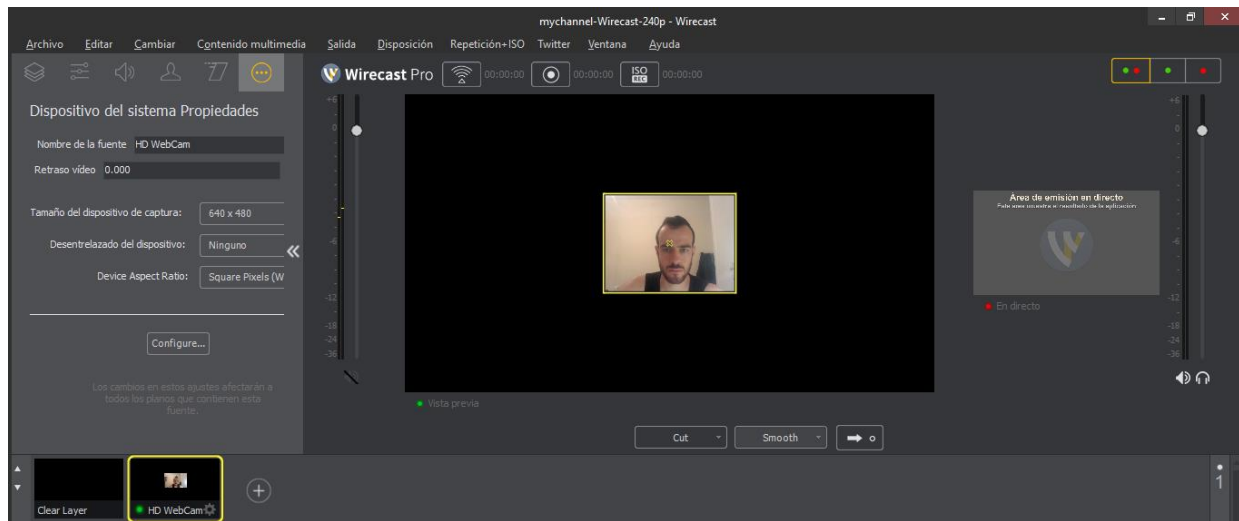


Ilustración 3.1: Pantalla principal Wirecast

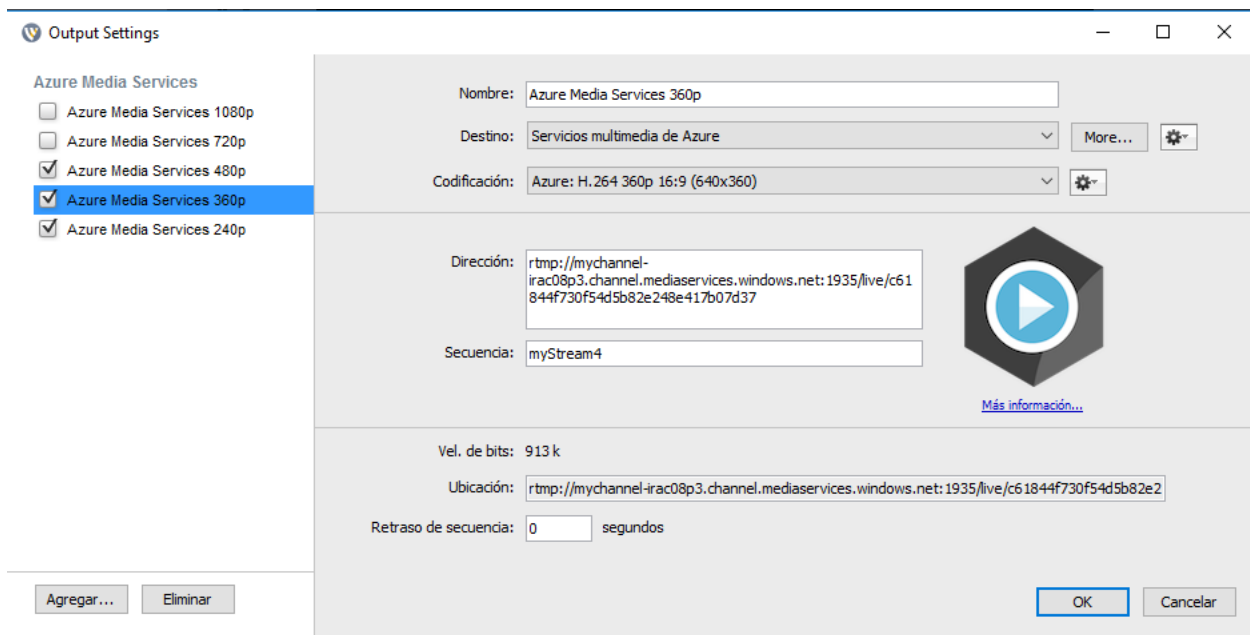


Ilustración 3.2: Configuración de salida Wirecast

3.2 Cliente Web

Hemos explicado que el DASH Industry Forum es un conglomerado de compañías que cooperan por usar este estándar con el propósito de evitar un mercado segmentado y un ofrecer una mayor interoperabilidad. Éste ofrece ciertas guías de implementación del estándar y, además, está desarrollando el proyecto dash.js [6]: aprovechando las APIs que ofrecen actualmente los navegadores (Media Source Extensions y Encrypted Media Extensions) definidas por el W3C, que permiten la decodificación y

reproducción adaptativa multiplataforma, han desarrollado un reproductor de código abierto basado en JavaScript como referencia para desarrollar para otras plataformas.

Para este proyecto, hemos hecho uso de este reproductor para desarrollar un simple cliente web que nos permita, por un lado, visualizar el streaming en vivo descrito anteriormente o cualquier contenido multimedia que permita format DASH, y por otro, conocer algunos parámetros básicos de la conexión a tiempo real, como la velocidad de bajada o el tamaño del buffer.

La integración del reproductor es tan sencilla como importar su fichero .js en el código HTML de la página web, como se especificó en el anterior capítulo:

```
<script type="text/javascript" src="http://cdn.dashjs.org/latest/dash.all.min.js"
defer></script>
```

Y posteriormente, inicializarlo en el cliente y asociarlo a una etiqueta <video></video> de HTML:

```
var url_streaming = "http://dash.edgesuite.net/akamai/bbb_30fps/bbb_30fps.mpd";
var player = dashjs.MediaPlayer().create();
player.initialize(document.querySelector("#streaming"), url_streaming, true);
```

Como se puede observar, solo requiere la dirección del fichero MPD de DASH para dar funcionalidad.

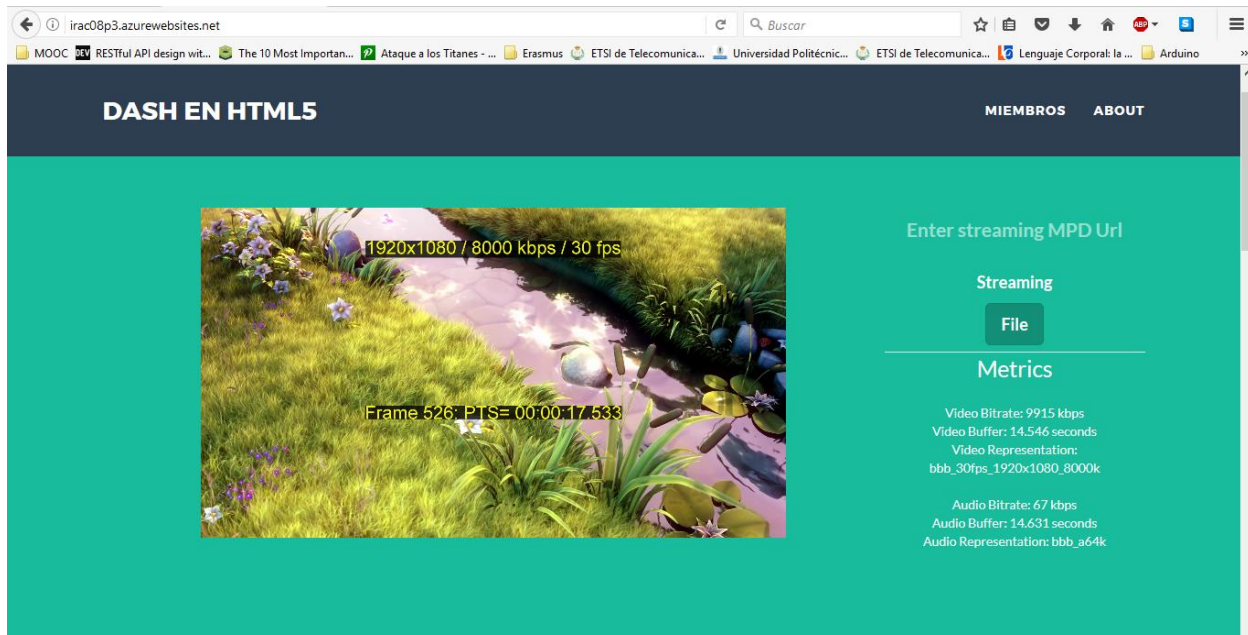


Ilustración 3.3: Cliente Web

Con el botón Streaming se carga el MPD de la dirección URL introducida y con el botón File se carga un vídeo preestablecido usado para depurar y comprobar funcionalidad.

El “look and feel” de nuestro proyecto se ha mejorado con el uso de una plantilla Bootstrap.

Por otra parte, monitorizar los parámetros ha sido relativamente más complejo, en el sentido de que, al tratarse de un proyecto en curso, la documentación no es muy exhaustiva [7]. Como prueba de concepto, monitorizamos, tanto de vídeo y audio, tres parámetros: la velocidad de transmisión, el tamaño del buffer y la representation, es decir, la codificación actual que se ha transmitido (no coincide con la de reproducción por el efecto del buffer).

El código del cliente se puede encontrar en el siguiente repositorio en GitHub:

<https://github.com/AntonioAlfrz/DASH-Web-Player>

Y lo hemos alojado en la nube y se puede encontrar en el siguiente enlace:

<https://irac08p3.azurewebsites.net/>

4. Conclusiones

En este trabajo principalmente hemos desarrollado un proyecto propio que aprovecha el trabajo realizado por el DASH-IF como implementación de DASH.

Partimos de una base teórica sobre DASH, en la que hemos investigado el funcionamiento de esta técnica de streaming de contenido multimedia. Contamos el modelo basado en el control total desde el lado del cliente, que va pidiendo los recursos necesarios a través del fichero XML MPD (Media Presentation Description), el cuál describimos en un subapartado. Especificamos su estructura general y una breve descripción de cada campo, junto con su propósito.

Hemos visto que existen diversas implementaciones de la técnica DASH, de muchas empresas y grandes multinacionales, y que está integrado en servicios famosos como Youtube o Netflix, que implementan MPEG-DASH. Sin embargo, concluimos que DASH-IF Reference Player es la iniciativa con más interoperabilidad y por tanto la que usamos para nuestro proyecto propio.

Finalmente, en cuanto a nuestro proyecto, hemos desplegado una pequeña plataforma de streaming sobre Azure, haciendo un estudio previo de las posibilidades y el alcance de los recursos disponibles, como el uso de Telestream Wirecast para la generación de contenido fácilmente integrable en Azure. Desarrollamos un sencillo cliente web con las guidelines aportadas por el DASH-IF, unidos a una plantilla de Bootstrap para complementar el frontend de nuestra solución.

5. Bibliografía

- [1] «Dynamic Adaptive Streaming over HTTP,» Wikipedia, [En línea]. Available:
https://en.wikipedia.org/wiki/Dynamic_Adaptive_Streaming_over_HTTP.
- [2] B. C. Núñez, «DASH: Un estándar MPEG para,» Universitat Politècnica de Catalunya.
- [3] DASH-IF, «Dash-Industry-Forum/dash.js Wiki · GitHub,» [En línea]. Available:
<https://github.com/Dash-Industry-Forum/dash.js/wiki>.
- [4] Microsoft, «Documentación Streaming sobre Azure Media Services,» [En línea]. Available:
<https://docs.microsoft.com/es-es/azure/media-services/media-services-overview#a-namea-idlivescenariosadelivering-live-streaming-events-with-azure-media-servicesa-idlivescenariosaentrega-de-eventos-de-streaming-en-vivo-con-servicios-multimedia-de-azure>.
- [5] Microsoft, «Codificadores locales para Azure Media Services,» [En línea]. Available:
<https://azure.microsoft.com/es-es/blog/azure-media-services-rtmp-support-and-live-encoders/>.
- [6] D. I. Forum, «Proyecto dash.js,» [En línea]. Available: <https://github.com/Dash-Industry-Forum/dash.js/>.
- [7] D. I. Forum, «Documentación dash.js,» [En línea]. Available:
<http://cdn.dashjs.org/latest/jsdoc/module-MediaPlayer.html>.
- [8] [En línea]. Available: https://en.wikipedia.org/wiki/Dynamic_Adaptive_Streaming_over_HTTP.