W3C Web and TV Workshop, 8-9 February 2011

Adaptive HTTP streaming and HTML5

Mark Watson, Netflix Inc.

1 Introduction

1.1 Netflix background

Netflix is a leading provider of streaming video services in the US and Canada. We offer a service whereby subscribers can access 10,000s of movies and TV shows on-demand over the Internet for a low monthly subscription fee. Our streaming service was launched in 2007 and currently we delivered over twice as much content over the Internet than through our successful DVD-by-mail service. In late 2010, Sandvine estimated that 20% of US peak time internet traffic was from Netflix [1].

The service is available on a wide variety of devices: Windows and Mac OS computers, game consoles, set-top-boxes, BluRay players, televisions, mobile phones and a well-known tablet device.

Our proprietary streaming technology is based on the standard HTTP protocol. Content is delivered using several well-known Content Delivery Networks (CDNs) and our website, database, security and other services deployed largely on Amazon Web Services.

All our content is protected using Digital Rights Management technology, to ensure that the content is played back only on devices which meet security requirements approved by the content providers. Choice of DRM technology is determined by the end devices, so we must support multiple technologies.

Our user interfaces are increasingly implemented in HTML5 and in December we announced our desire to see open standards for adaptive streaming integrated into HTML5 [2].

1.2 The need for standards

Today, Netflix provides an SDK which must be integrated into a device in order to provide support for the Netflix service. We test and certify devices to ensure the SDK has been correctly integrated and that the device meets security requirements. This limits the number and types of devices on which our service can be made available.

A standard for adaptive streaming integrated into HTML5 would eliminate the need for service-specific software integration and so would be of great advantage both to device

manufacturers – who could more easily increase the number of services on their devices – and to service providers like ourselves, who could gain access to more devices.

This paper outlines our view on the requirements for such standards and the progress to date in other standards bodies such as MPEG and IETF.

2 Adaptive HTTP Streaming

High quality video streaming on the Internet requires adaptivity: that is, the service must adapt to the available bandwidth, which can vary significantly during a viewing session. We do not believe that network or transport layer technologies (e.g. Network Quality of Service) will change this fact any time soon. We believe Adaptive HTTP Streaming is the technology of choice for video streaming on the Internet.

2.1 General model and adaptive streaming manifests

At its core, adaptive HTTP streaming implies advertising to a client a set of available streams and having the client make choices as to which part of which stream to download when. Streams must be accurately time aligned to enable seamless switching.

The data in an adaptive streaming system can be modeled in two layers:

- The "manifest" containing information about the available streams
- The streams themselves

The information needed at the "manifest" layer is *just that needed to make the choices of which streams to select*. The information at the stream layer is *that needed to access and play the media once it has been selected*.

Figure 1 illustrates a general model for the manifest layer for on-demand services. Getting the general model right is important, whereas the specific encoding format for the manifest (e.g. XML, M3U8) is not.

In this model, *tracks* represent different time-aligned media associated with a presentation – for example audio, video, subtitles. *Streams* within a track represent alternative encodings of the exact same source media.

It is essential for services such as ours to store separate media types separately. The alternative (multiplexing audio and video into a single file) leads to a combinatorial explosion of streams as soon as multiple audio tracks (e.g. multiple languages) are considered.

In addition, it is essential that streams are stored unchunked. Splitting a stream into small chunks in time (i.e.

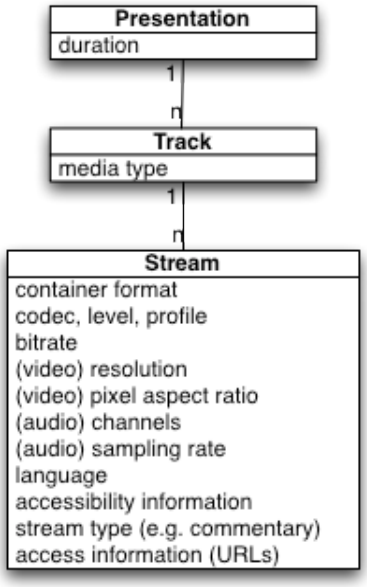

Figure 1: Streaming Manifest Model

10 seconds) results in billions of files, and does not scale or cache well.

Finally, it is valuable to include redundancy by storing each file at multiple locations (specifically in multiple CDNs.) This results in a data model where each stream has a set of URLs all pointing to an identical file containing the entire stream.

The MPEG DASH draft defines an XML manifest format including support for the model and features above. We believe this forms a good basis for a common adaptive streaming manifest format. We propose definition of a simplified profile of the MPEG DASH standard for on-demand streaming.

In IETF there has been no progress on standardizing adaptive streaming and in fact a proposal to establish an 'http streaming' working group has not received substantial support.

2.2 Media container formats

To be useful for on-demand adaptive HTTP streaming, a media container needs to store the media "progressively" – that is, all the information needed to playback any given portion in time of the stream should be gathered together in one place in the file.

Fragmented MP4 files and WebM files both meet this requirement (we use the former in our service).

As noted above, some form of index information is needed to enable the client to form partial HTTP requests. Specifically the index needs to map *time ranges* to *byte ranges*. MPEG is defining a new Segment Index Box within the ISO Base Media File Format (on which MP4 is based) for exactly this purpose. In WebM files the Cue data could fulfill this same purpose.

3 Digital Rights Management

Digital Rights Management is an essential component for delivering high quality commercial content over the Internet.

At Netflix, we support multiple DRM types, as appropriate for a specific device that we are streaming to. We do NOT suggest that W3C should standardize DRM technologies themselves. However, by standardizing on an encryption model, the role of the DRM System is limited to usage rights and key acquisition, and it becomes possible for a single file to be decrypted by multiple DRM providers.

3.1 Common Encryption

Several major DRM vendors have agreed to a common approach to encryption under the auspices of the DECE consortium, based on the Microsoft PIFF specification. Introduction of

this solution into the ISO Base Media File Format is being discussed. This enables a single file to be decrypted by clients supporting different DRM technologies.

3.2 Common Authentication and Authorization

Service authorization is a service-specific function. In the case of Netflix we must check that the user is a valid subscriber, that access to the content is within the terms of their subscription and that the particular device is authorized to view the content (for example HD content cannot be viewed on devices with weaker DRM implementations). This implies that the service must authenticate both the user and the device.

Device authentication requires access to DRM-independent device credentials which are securely bound to the physical device and the ability to construct cryptographic proof of possession of those credentials.

Having authorized a user to view a given piece of content on a given device, we can apply the DRM technology to ensure the content is available only to that device. This requires a secure binding between the DRM license transaction and the device identity.

We believe there is a need to standardize some basic tools, using well-known techniques, for managing secure device identity and binding this to the DRM operations such as license transaction in a DRM-independent way. This will enable development of services which are largely DRM-technology-independent which will be of great advantage to the industry as a whole.

4 HTML5 Integration

We believe that HTML5 should support a standard adaptive HTTP streaming protocol. This implies the following:

- Defining a standard manifest
- Providing support on the HTML5 media tags for track advertisement and selection
- Expose the additional states, events and parameters that exist in an adaptive streaming context on the media tags
- Providing support for security and DRM integration

4.1 Manifest format

Adaptive HTTP streaming also exists outside the context of HTML5, so it is desirable to agree on a manifest format which is not specific to HTML. The obvious approach is that a URL for a manifest file can be provided in the src attribute or <source> element of the <video> element.

4.2 Track advertisement and selection

A natural aspect of adaptive HTTP streaming is that many alternative time-aligned tracks may be available for a single content item, particularly including tracks providing different languages and providing for accessibility needs. Adaptive streaming particularly introduces the requirement for strict time alignment and the possibility of simple and seamless switching.

It is therefore necessary to have a way to advertise on the <video> tag the available tracks and provide controls to enable/disable tracks before and during playback.

4.3 Additional states, events and parameters

Adaptive streaming introduces the possibility of stream change events, which should be reported to the application. There may also be other state/event implications. Introduction of multiple URLs for the same file introduces some requirements for reporting on URL choices to support service level logging.

Many services, including ours, collect extensive quality metrics on user streaming sessions. MPEG DASH is working on high level definition of such metrics and integration into HTML5 should ensure that enough information is exposed to the application to report such metrics.

Adaptive streaming introduces an additional element of complexity in terms of the algorithms and heuristics. There needs to be a means for the algorithms and parameters to evolve. We believe that a pluggable heuristics engine and dynamic parameter provisioning are essential.

4.4 Security and Digital Rights Management

As described above, a standard method to integrate DRM technologies with HTML5 streaming is needed. It should be possible to provide features such as authentication and authorization in a service-specific way, using standard tools, rather than a proprietary DRM-technology-specific way.

5 Conclusions

In this paper we describe a number of areas where we believe additional standardization work is required to ensure that HTML5 can become a viable platform for commercial video services such as that offered by Netflix.

Specifically, we believe there is a need for:

- Agreement on a standard adaptive streaming model and manifest format
    - *We believe the MPEG DASH standard, or a simple profile thereof, fulfills the requirements for this*
- Agreement on one or more standard media container formats, and required features/profiles for adaptive streaming

    - *Fragmented MP4 files, using the latest ISO Base Media File Format draft amendment, fulfill the requirements*
    - *Some work is required to define how WebM files can be used in this context, but it is likely that the existing data structures can effectively support adaptive streaming*
- Extension of the HTML5 media elements to support a defined adaptive streaming manifest format, track advertisement and selection and additional states, events and parameters required for quality metric collection and experimentation with adaptivity heuristics
- Definition of a standard for DRM integration with HTML5, including simple tools for DRM-independent device authentication and authorization

References:

[1] http://www.sandvine.com/news/global_broadband_trends.asp
[2] http://techblog.netflix.com/2010/12/html5-and-video-streaming.html