HTML5: Now with Premium Content

Bent G Christensen, Ken Morse, Ajith Nair, Bill Ver Steeg
Cisco Systems, Inc.
{benchris, drken, naira, versteb}@cisco.com

August 15, 2011

1 Introduction

In the realm of IP video delivery there has traditionally been a number of different architectural approaches, e.g., for managed enterprise and service provider networks compared to unmanaged networks for over-the-top services such as YouTube, Hulu, iPlayer, and Netflix. Both of these architectural approaches offer different benefits and drawbacks. Cisco's video expertise and experience has mainly been with end-to-end solutions in managed networks for sustained delivery of high quality and premium content in a protected environment. But there has also been an increased interest in supporting personal and mobile devices for apps and content playback, and this is where HTML5 fits in very well.

So, with this perspective and Cisco's background in mind the following topics are areas where we see HTML5 need some more standardization and development in order to become the ideal platform for media consumption and interactive applications for unmanaged as well as managed networks.

2 HTTP adaptive streaming

For robust delivery of content especially over the Internet or unmanaged home networks, the streaming of video should be able to adapt to the available band-

width during a viewing session. Most adaptive bit-rate streaming protocols are divided into two layers:

- A manifest file providing metadata of the content streams available, and
- the actual content streams encoded in different bitrates or qualities.

The manifest file allows the client to select the most appropriate content stream given the bandwidth available and device decoding capabilities. These two layers are not directly dependant and can be defined separately, i.e., the manifest file format could be standardized where as the content could be contained and or encoded with proprietary mechanisms if needed.

For this the Dynamic Adaptive Streaming over HTTP (DASH) efforts look promising. Especially the MPEG DASH for efficient delivery of MPEG media over HTTP, which is influenced by the 3GPP AHS specification.

Firstly the use of HTTP fits well with clients already supporting HTML5, and HTTP-based delivery can easily use existing network paths through NAT and firewalls without the need for re-configuration.

Secondly, the MPEG formats are the de facto standard for high quality content encoding. The underlying data flows enable a wide range of functionality, from lip synch to stream splicing to rate adaptation. There is a rich set of user experiences available, from Video on Demand style viewing to watching live events. There are some challenges to be addresses as this technology is driven to the mass market, and touch points between MPEG, 3GPP, W3C and IETF should be carefully considered.

3 Content protection

To allow premium content through HTML5 apps most Content Providers require some form of content protection or DRM mechanism to protect their content. All of these protection schemes are proprietary and it does not fit well with the W3C charter to dictate a specific and possibly proprietary mechanism for all premium content. Instead it would make sense for the HTML5 effort to facilitate different DRMs through a framework or plug-in architecture with a simple DOM API extension such as:

- `isContentPlayable(url content)` or more informative
- `getContentConstraints(url content)`.

This DRM framework architecture should promote decoupling between transport mechanism and the DRM scheme used to enhance flexibility.

The UltraViolet (UV) digital rights and authentication system seems like an appropriate first candidate and a good recommendation of a DRM without dictating it from HTML5. Most of the major film studios as well as many of the smaller ones support UV. UV can for instance be delivered through MPEG DASH as mentioned previously.

4 Metadata and companion devices

Besides actual video content metadata is also required for the user the search, browse and find the content they need in a HTML5 context. This metadata includes:

- TV Guide data,
- Video-on-demand catalogs,
- Movie details: year, length, actors, ...
- and more

Most of this metadata can be fetched and delivered through existing and proven W3C mechanism such as backend Web services using RESTful API to clients using AJAX or WebSockets. The metadata can then be presented and navigated by the user through traditional Web technologies, e.g., JavaScript and CSS.

An interesting development is the demand for more social interaction between users and thereby also between their different devices. There have been many efforts to accomplish this for instance Facebook or Tweeter apps layered on top of video content being displayed. But also being able to remote control one device from another companion device is becoming a more common demand.

For this the XMPP communication standard (aka Jabber) looks like a perfect fit to: securely login, exchange capabilities, send commands between devices, and convey presence information.

5 Profiles for compliance and performance

For implementers of an HTML5 based media player a set of profiles defining the core set of APIs and the performance requirements or expectations for each of these would be useful. This not only provides a common test harness for a certain profile, but also provides a framework for comparing different implementations and thereby creating focus on optimization and performance.

All the new software requirements discussed in this paper requires a new analysis on the processing power required for the CPU in any implementation, whether it be a television, tablet, STB or mobile phone. The days of a SoC that had a MIPS or ARM CPU with about 300 Dhrystone MIPS of CPU processing capability is history. In most non-pc hardware implementations many of the functions such as video decoding, encoding in some cases and graphics have hardware acceleration modules. However, many of the acceleration modules are limited in their capabilities because of the silicon required and hence pushing the overall cost of the devices. So the new paradigm calls for hardware assist, but also very powerful CPUs with DMIPS in the range of 4000 to 10000. This would also allow for supporting even new codecs that have not been included in the hardware assist modules directly.

6 Conclusion

In this paper we described a number of different topics that we suggest more standardization work is required in order to improve HTML5 for premium content consumption.

Our recommendations for HTML5:

- For an adaptive bitrate streaming protocol we recommend adopting MPEG DASH.
- Extending the HTML5 APIs with simple DRMs calls, and use UV as a test case DRM.
- To retrieve metadata and control messages between client devices we recommend XMPP as the communication protocol.

- And finally, a set of HTML5 profiles with a corresponding test-harness would benefit implementers and provide a common framework for performance comparison.