Synchronisation standardised for connected TVs

Matt Hammond
BBC Research and Development

matt.hammond@bbc.co.uk
This presentation

1. The Journey

2. Standards

3. Challenges
1. The Journey

Ideas we tried

... and things we learnt
Play along or companion content with broadcast

Audio watermarking  
... robustness?

IP push triggers  
... accuracy?

... time-shifted viewing?
Experiences valued most when content was specific to the episode and supported interests.

Users want content to be synchronised …

… but still have the freedom to browse / navigate
“Universal Control API”

Motivation: improve UI accessibility

- Expose listing of available content
  - Broadcast and on-demand, unified

- Control of TV functions
  - Volume, pause, subtitles,
  - Content selection, Recording
  - Power / standby

- Synchronisation
- Content Identification

http://www.bbc.co.uk/rd/publications/whitepaper194
Universal Control API

- Commercially unacceptable to manufacturers …
  - Takes away too much control of the UX

- … so a middle ground needed

- Useful lessons learnt:
  - **Content identification and synchronisation** core to many use cases
  - **Decoupling** of the companion experience from the TV gives flexibility
2. Standards

HbbTV 2.0

… the parts relevant to synchronisation

hbbtv.org
Standards for TV technologies

• The TV is an appliance
  – “Just Works”
  – Feature set is not updated
  – Long lifetime
  – Expectation of simple to use and consistent experience

• For broadcasters…
  – Very well specified behaviours that are consistent between manufacturers
What is HbbTV?

- Open spec agreed with major TV manufacturers for the European market
- Defines an HTML+JS interactive application environment on the TV
- Single app running at any time
- HbbTV 2.0 major update
  - Developed Oct 2012 - Feb 2015
  - HTML5, synchronisation, companions, MPEG DASH etc…

![HbbTV Image]
Synchronisation in HbbTV 2.0

Multi-stream synchronisation

Inter-device synchronisation

DVB Broadcast

DVB “CSS” protocols

Synchronisation model and protocols defined by DVB
HbbTV Architecture for synchronisation

Defines a Javascript API

... and how it maps to protocols and media object behaviours
2. Standards

DVB “CSS” (“Companion Screens and Streams”)
… the parts relevant to synchronisation

*ETSI TS 103 286 (part 2)*
Problem domain: Many use cases and challenging requirements

- **Wide range of use cases:**
  - Guide to the channel you are watching
  - Information about an part of the programme
  - Companion content for adverts/trailers
  - Play along experiences
  - Access services (e.g. AD) on companion
  - Alternative camera angles

- **Minimal changes to existing broadcast signalling**

- **Work for IP delivered TV as well as broadcast TV**
Problem domain: Differing data models

- Different data models with entities representing different concepts for content IDs, timelines etc
  - Standardised broadcast metadata (e.g. DVB SI)
    But still some differences between platforms.
  - IP delivered services use many bespoke metadata models (e.g. bbc.co.uk/ontologies/po)
- Cannot force broadcasters to completely redesign their systems.
DVB CSS – The Approach

- TV tells companion: “I am showing this item of content”
  “This is where I am in the content”

- Let the companion decide what experience to create
  - TV does not “control” the experience on the companion

- Separation of companion content from TV platform specific mapping metadata
  - Metadata to map between broadcast world and online world data models
Content IDs and timelines that the HbbTV terminal support:

<table>
<thead>
<tr>
<th></th>
<th>Broadcast</th>
<th>Broadband</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Content ID</strong></td>
<td>MPEG TS DVB locator URL</td>
<td>MPEG TS URL of stream</td>
</tr>
<tr>
<td><strong>Possible Timelines</strong></td>
<td>PTS or MPEG TEMI</td>
<td>PTS Composition Time</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Time since start of a Period</td>
</tr>
</tbody>
</table>

But can be easily extended in future specs / standards:

- **Content IDs are URIs**
- **Timelines are identified by URNs and their tick rate can be arbitrary.**
The protocols between TV and companion

Broadcast or Internet TV

HbbTV Terminal

DVB Synchronisation protocols

Content IDs

Timestamps

Clock Synchronisation

Companion App

Companion Content

Mapping Metadata
The protocols between TV and companion

Broadcast or Internet TV

**HbbTV Terminal**

**Companion App**

**DVB Synchronisation protocols**

**CSS-CII:** Content Identification

*WebSockets + JSON*

**CSS-TS:** Timeline Synchronisation

*WebSockets + JSON*

**CSS-WC:** Wall Clock Synchronisation

*UDP*

**CSS-MRS**
The protocols – CSS-CII (WebSockets+JSON) … Content Identification

1. Companion connects
2. TV start sending JSON “CII” messages:

```json
{
    ...
    "contentId" : "dvb://233a.1004.1044;363a~20130218T0915Z--PT00H45M",
    "contentIdStatus" : "partial",
    "presentationStatus" : "okay",
    "wcUrl" : "udp://192.168.1.5:5800",
    "tsUrl" : "ws://192.168.1.5:5815"
    ...
}
```

(just an example - real message may contain different combinations of properties)

Would be URL of the stream/manifest if TV was playing internet video.
The protocols – **CSS-WC** (UDP) … Clock synchronisation

1. Companion regularly sends a request. TV sends a response. t1, t2, t3, t4 recorded

2. Companion uses this measurement to calculate the clock time **difference** and adjust

   - Companion also calculates **dispersion** (error bounds)
     - Combines round-trip time and potential for slew due to oscillator frequency error
     - Increases … until a new lower-dispersion measurement taken

*Inspired by NTP, but much simplified.*

\[
\text{Difference} = \frac{(t2+t3)}{2} - \frac{(t1+t4)}{2}
\]
The protocols—CSS TS (WebSockets+JSON) … Timeline Synchronisation

TV is “master” and controls the timing

Companion can also send timestamps back. TV is free to choose to ignore or utilise them.

Multiple companions can synchronise to the one TV

- Control Timestamps
- Actual Presentation Timestamps
- Earliest Presentation Timestamps
- Latest Presentation Timestamps
The protocols– CSS TS (WebSockets+JSON) … Timeline Synchronisation

1. Companion connects
2. Companion sends “Setup Data”

```json
{
    "contentIdStem" : "dvb://233a.1004.1044",
    "timelineSelector" : "dvb:css:timeline:pts"
}
```

3. TV responds with “Control Timestamps”

```json
{
    "contentTime" : "1200",
    "wallClockTime" : "987654000000",
    "timelineSpeedMultiplier" : 1.0
}
```

(...or a timestamp indicating “timeline is unavailable”)

Actual, Earliest, Latest Presentation Timestamps not shown here
The protocols – CSS-MRS (HTTP+JSON) … Mapping information

- Content ID used as key to retrieve “Material Information” from MRS

- Platform differences encoded as mappings from content IDs and timeline intervals to Materials with timelines

- Companion does not need any platform specific knowledge
The protocols – CSS-MRS (HTTP+JSON) … Mapping information

1. Companion makes HTTP request to the Material Resolution Service (MRS)
   Content ID from TV is part of the URL

2. JSON Response describes mappings
   “Material Information” data model

   ```json
   {
     "type" : "response",
     "version" : "1.1",
     "rev" : "20130815095215",
     "repollInterval" : 30,
     ...
     "materials" : [ ... ],
     "syncTimelineInformation" : [ ... ]
   }
   ```

   Each **Material** represents something meaningful to the companion
   (channel, programme, trailer, segment, etc).

   How to **map** from TV timeline positions to Material timelines

   can also carry links to “update” services
The protocols – CSS-MRS (HTTP+JSON) … Mapping information

A “Material”

```
{
    "index" : "1bcded38994",
    "parents" : [ ],
    "ids" : [ { "type" : "tag:bbc.co.uk/programmes/episode", "id" : "b00cr16g" } ],
    "timelineProperties" : { "unitsPerTick" : 1, "unitsPerSecond" : 1000 }
}
```

Companion recognises Materials by the ID(s). Any type allowed. Ignore unrecognised ones.

Does not define its own concepts of programmes, adverts etc

Instead, use IDs to relate a Material to entities in existing data models
The protocols – CSS-MRS (HTTP+JSON) … Mapping information

Timeline to request from the TV …

```
{  
    "contentIdStem" : "dvb://233a.1004.1044",
    "timelineSelector" : "urn:dvb:css:timeline:pts",
    "timelineProperties" : { "unitsPerTick" : 1, "unitsPerSecond" : 90000 },
    "mappings" : [  
        {  
            "materialIndex" : "1bcd38994",
            "start" : "297000",
            "end" : "321000",
            "correlationsChanging" : false,
            "correlations" : [  
                { "point" : "297000", "materialPoint" : "0" }
            ]
        }
    ]
}
```

… ask for a timeline based on PTS

When content ID begins with this “stem”

… and mappings to the Material’s timeline
The protocols between TV and companion

Broadcast or Internet TV

HbbTV Terminal

DVB Synchronisation protocols

- **CSS-CII**: Content Identification
  - WebSockets + JSON
- **CSS-TS**: Timeline Synchronisation
  - WebSockets + JSON
- **CSS-WC**: Wall Clock Synchronisation
  - UDP

Companion Content

Companion App

CSS-MRS
3. Challenges

... some thoughts from a broadcaster perspective
Challenges ... latency

Coding, Mux & Modulation

~ 5 seconds

Coding & Segmentation

Segment buffering

> 10 seconds
Challenges – how best to estimate and adjust audio/video timing?

Algorithms to maintain accurate clock and presentation sync while minimising disturbance

Does the best strategy depend on the type/genre of content?
Challenge ... accommodating a messy experience lifecycle

The user will switch attention, activities, channels ...

... how does a synchronised experience accommodate this?

Can I still play along if I join mid-way through the quiz, or miss question 3?
Challenge ... leveraging atomised media / object based broadcasting

Atoms of media composed responsively at the TV / companion

Use these elements to create synchronised multi-device experiences?
Challenge … multi-device and IoT

New multi-device synchronised heterogeneous content possibilities

Internet of Things:

• New types of “media” to synchronise (light bulb colour?)

http://bbc.in/1IdR56o
Challenge ... more standardisation!

TV is a complex technology environment, and becoming more complex as we collide with Web technologies.

• Getting harder to create services that work across lots of platforms

Standardisation, common APIs, common frameworks enable services

• Unifying different models and protocols under common APIs (W3C?)
• Tools to publish across multiple platforms
Thank you!

(And please come and see the demo!)