Marrying WebRTC and DASH for Interactive Streaming

IETF 113, 21 March 2022
## DASH + WebRTC Use Cases

### Fallback
Real-time WebRTC with fallback to DASH
- Device doesn't support WebRTC.
- Network connection is not good enough to sustain a very low latency stream.
- Premium real-time experience using WebRTC not enabled for this user.

### Interleaved
Real-time WebRTC interleaved with DASH
- A real-time live event using WebRTC while the ad periods are delivered with DASH.
- Main content delivered via DASH and periods for interactive programs delivered via WebRTC.
- A real-time event using WebRTC with supplemental pre-recorded content delivered with DASH.

### Concurrent
Real-time WebRTC concurrent with DASH
- Pre-recorded DASH content delivered with supplemental live WebRTC streams.
- Co-watching synchronized streams with audio/video/text chat.

### Device doesn't support WebRTC.

### Network connection is not good enough to sustain a very low latency stream.

### Premium real-time experience using WebRTC not enabled for this user.

### A real-time live event using WebRTC while the ad periods are delivered with DASH.

### Main content delivered via DASH and periods for interactive programs delivered via WebRTC.

### A real-time event using WebRTC with supplemental pre-recorded content delivered with DASH.

### Pre-recorded DASH content delivered with supplemental live WebRTC streams.

### Co-watching synchronized streams with audio/video/text chat.
## Comparing DASH and WebRTC

<table>
<thead>
<tr>
<th>Feature</th>
<th>DASH</th>
<th>WebRTC</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Content Description</strong></td>
<td>MPD: Same for all users, describes all available content</td>
<td>SDP: Unique for each client, typically with a single audio and a single video.</td>
</tr>
<tr>
<td><strong>Media Selection</strong></td>
<td>Client selects media, bitrate, and codecs.</td>
<td>Codecs negotiated between the server and client, server selects or adapts bitrate.</td>
</tr>
<tr>
<td><strong>Subtitles/Captions</strong></td>
<td>Standardized</td>
<td>Proprietary</td>
</tr>
<tr>
<td><strong>Timing</strong></td>
<td>Buffered and time synched</td>
<td>Immediately rendered</td>
</tr>
<tr>
<td><strong>End-to-end Latency</strong></td>
<td>~ 3-5 seconds for LL-DASH</td>
<td>&lt; 1 second, often less than 0.5 second</td>
</tr>
<tr>
<td><strong>Distribution</strong></td>
<td>CDN: low-cost, widely available</td>
<td>WebRTC servers: proprietary implementations of standards</td>
</tr>
<tr>
<td><strong>Playback</strong></td>
<td>JavaScript libraries</td>
<td>Direct support in all modern browsers</td>
</tr>
</tbody>
</table>
Hybrid Delivery

Publisher: Live Content

Publisher: Pre-recorded Content

Cloud Services and Storage

Content Processing (Transcoding, Packaging, etc.)

WebRTC Stream

Scaled Delivery

DASH Manifest and Chunks

CDN

Subscriber
Example Hybrid Client Architecture

Media Player

- DASH Player UI
  - Track Selection
  - Media Rendering
- DASH Controller
  - Control
  - MPD Parser
  - Segment Parser
  - HTTP Client
- DASH Server(s)
  - MPD
  - Segments

WebRTC Client

- WebRTC Client APIs
- Standard WebRTC Stack
- HTTP Client
- WebSocket

WebRTC Server

- WebRTC Session
- Session Management
- Events
Workflow

WebRTC Media Server  →  WebRTC Server  →  DASH Server  →  DASH Player  →  WebRTC Client

- Request MPD
- MPD with WebRTC adaptation set(s)
- Parse MPD and select tracks
- WebRTC Info
- SDP Negotiation
- Establish WebRTC connection
- Establish WebSocket connection
- Media data (video, audio) via WebRTC
- Events via WebSocket (subtitles, captions, etc.)
- Render media

WebRTC Info
From Discovery to Streaming

Current state of Real-Time Streaming

Vendor A - Proprietary Manifest  Proprietary Session Negotiation  Standard WebRTC Stream
Vendor B - Proprietary Manifest  Proprietary Session Negotiation
Vendor C - Proprietary Manifest  Proprietary Session Negotiation

Goal for Real-Time Streaming

Standard Manifest  Standard Session Negotiation  Standard WebRTC Stream
Work for WebRTC Extensions

- Define and select appropriate session management/signaling protocol
- Define control protocol for dynamic stream switching that does not require SDP renegotiation
- Continue development of methods for additional security of streams
- Define a standardized means to deliver subtitles, closed captions, and other events
- Continue development of a mechanism for time synchronization of timed metadata and DASH periods
- Collection of metrics and client metadata for WebRTC sessions and translation to existing metrics and client metadata, transmission via APIs
Work for DASH

- Determine APIs to be used between WebRTC clients and DASH clients
- Define how WebRTC information is represented in MPDs
- Determine whether DASH and WebRTC can both render to a single browser video element or switch between two video elements
- Support hybrid operations with WebRTC and DASH HTTP-based operations
Further Reading

DASH-IF Report Summary: https://dashif.org/webRTC/


Interest survey: https://forms.gle/Yy89GGeMsXYQixBZ6