

HTTP Live Streaming Update

Session 408

Roger Pantos

Core Media Engineer

These are confidential sessions—please refrain from streaming, blogging, or taking pictures

Outline

- Brief introduction to HTTP Live Streaming
- New features
- Best practices

HTTP Live Streaming Overview

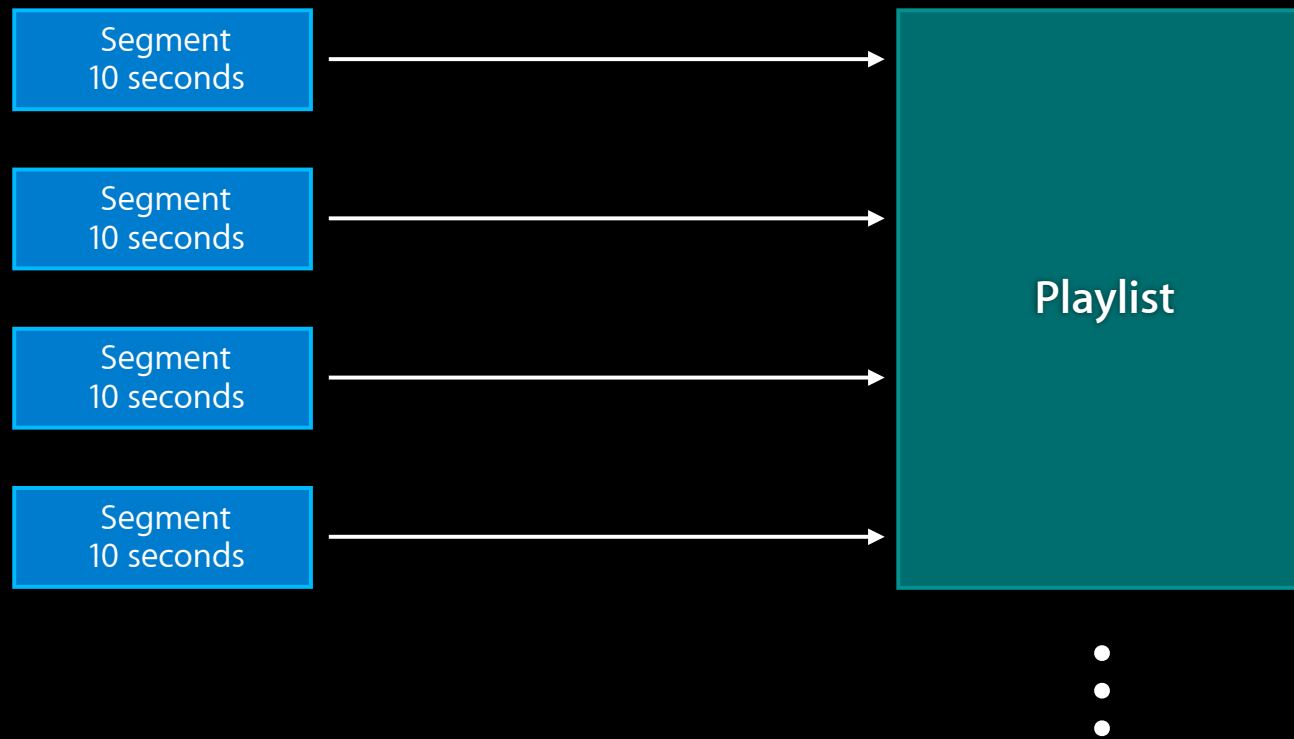
How HTTP Live Streaming Works

In 20 seconds!

- Break a media presentation up into segments (e.g. 10s each)
- Put a URL for each media segment into a Playlist file
- Put everything on an HTTP server

- Client downloads the playlist file
- Download and play each media segment
- Reload the playlist to discover new segments

Segments and Playlists



Video on Demand Playlist

#EXTM3U

#EXT-X-VERSION:3

#EXT-X-TARGETDURATION:10

#EXT-X-MEDIA-SEQUENCE:0

#EXTINF:9.7,

<http://example.com/movie1/fileSequenceA.ts>

#EXTINF:9.2,

<http://example.com/movie1/fileSequenceB.ts>

#EXTINF:8.6,

<http://example.com/movie1/fileSequenceC.ts>

#EXTINF:9.7,

<http://example.com/movie1/fileSequenceD.ts>

#EXT-X-ENDLIST

Some Refinements

- Server can offer several Playlists at different bit rates
 - Client will dynamically switch to adapt to network speed

Variant Playlist

```
#EXTM3U
```

```
#EXT-X-STREAM-INF:PROGRAM-ID=1,BANDWIDTH=150000  
http://example.com/low/index.m3u8
```

```
#EXT-X-STREAM-INF:PROGRAM-ID=1,BANDWIDTH=640000  
http://example.com/high/index.m3u8
```


Some Refinements

- Server can offer several Playlists at different bit rates
 - Client will dynamically switch to adapt to network speed
- A Playlist can contain discontinuities
 - Ad breaks, program boundaries
 - #EXT-X-DISCONTINUITY

Some Refinements

- Server can offer several Playlists at different bit rates
 - Client will dynamically switch to adapt to network speed
- A Playlist can contain discontinuities
 - Program boundaries, ad breaks
 - `#EXT-X-DISCONTINUITY`
- Media can be encrypted
 - Keys are specified in the Playlist
 - `#EXT-X-KEY:METHOD=AES-128,URI="https://example.com/key.php"`

Playback Control

- iOS
 - HTML5 <video> element
 - MediaPlayer: MPMoviePlayerController
 - AVFoundation: AVPlayerItem
- OS X
 - HTML5 <video> element
 - AVFoundation: AVPlayerItem

New Features

Alternative Audio

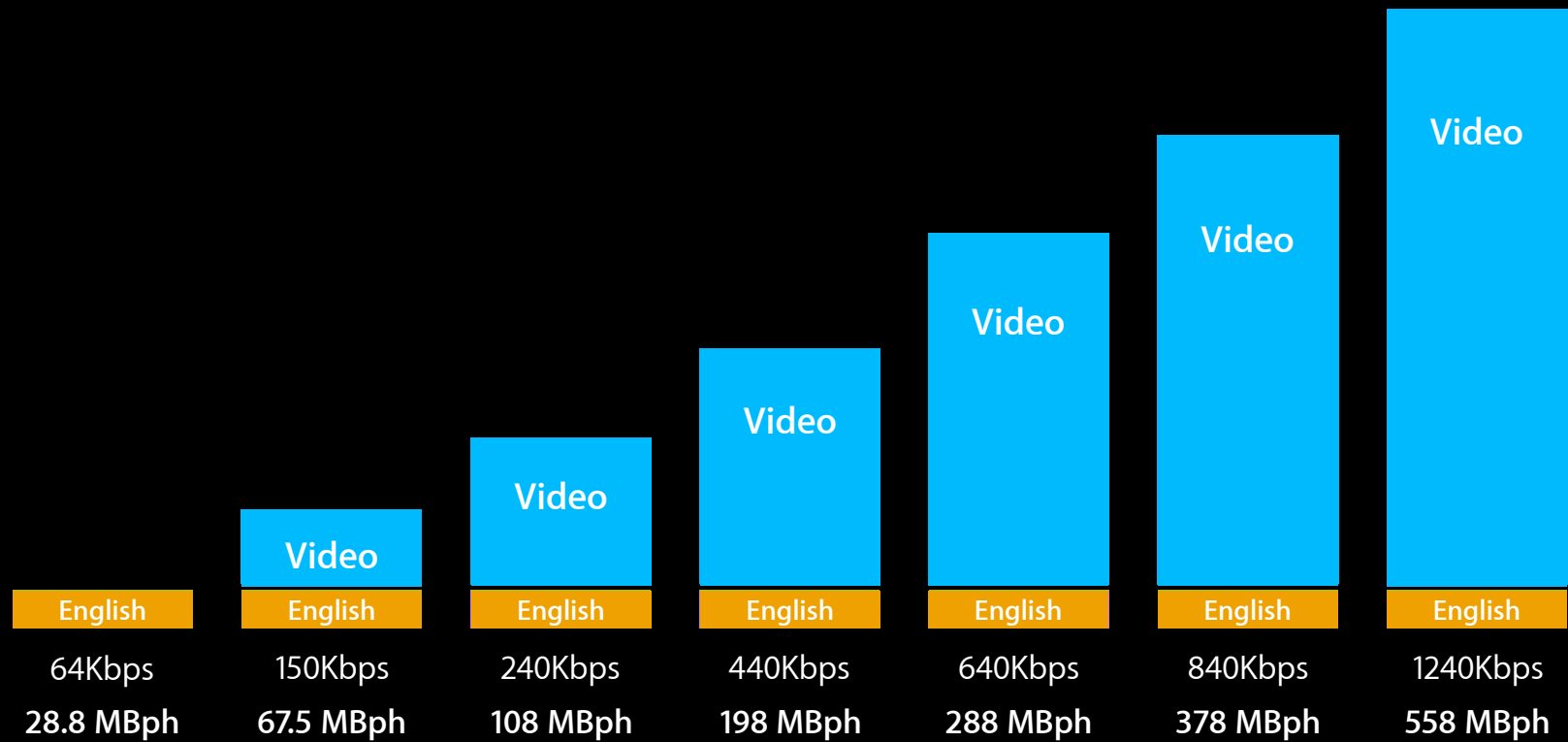
- Different audio can often accompany the same video
 - Other languages
 - Director's Commentary
 - Home/Away game broadcasts

Alternative Audio

- Different audio can often accompany the same video
 - Translations into other languages
 - Director's Commentary
 - Home/Away game broadcasts
- Challenges:
 - Downloading one stream with multiple audio tracks is inefficient
 - Duplicating video for each alternative audio is expensive

Baseline Storage

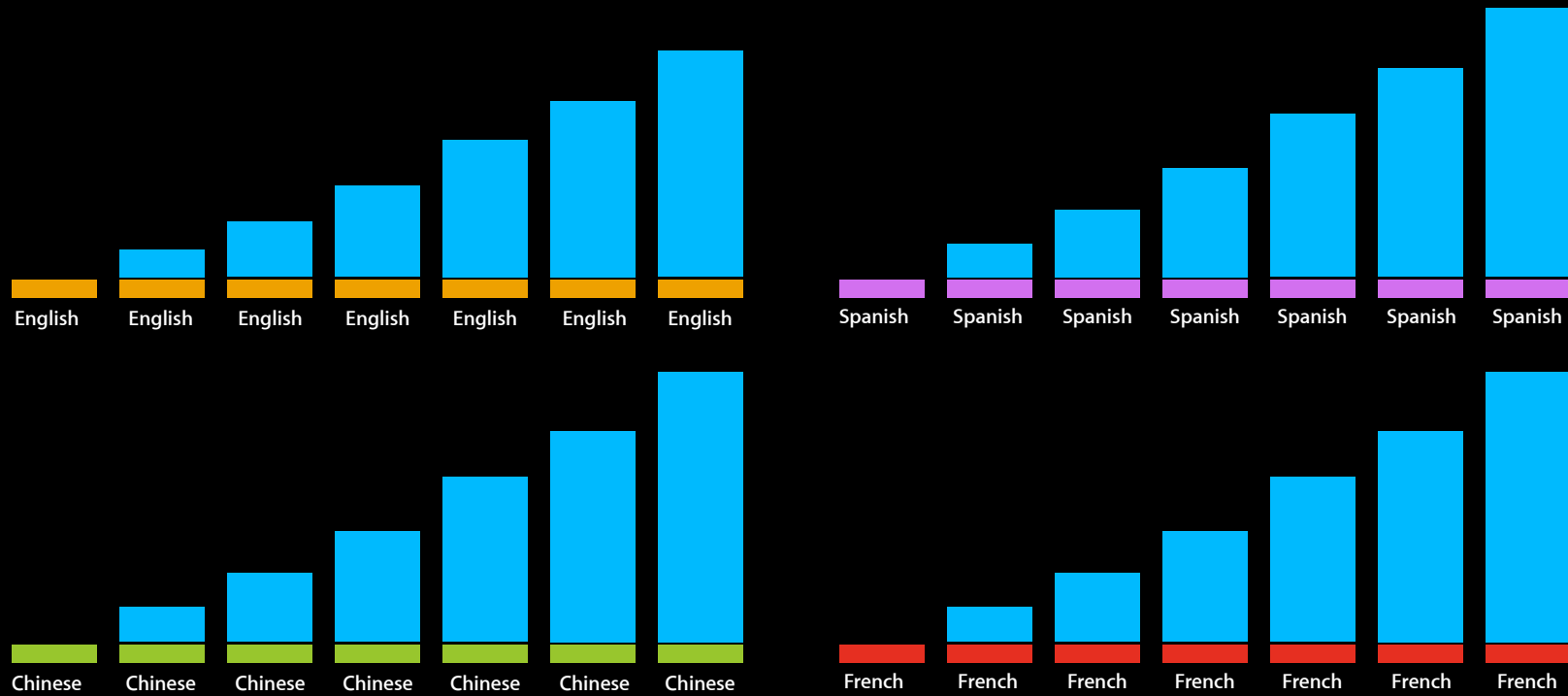
Video + 1 audio at recommended bit rates



Total: 1.626 GB per Hour

Duplicating Video

1 copy of video with 4 audio translations



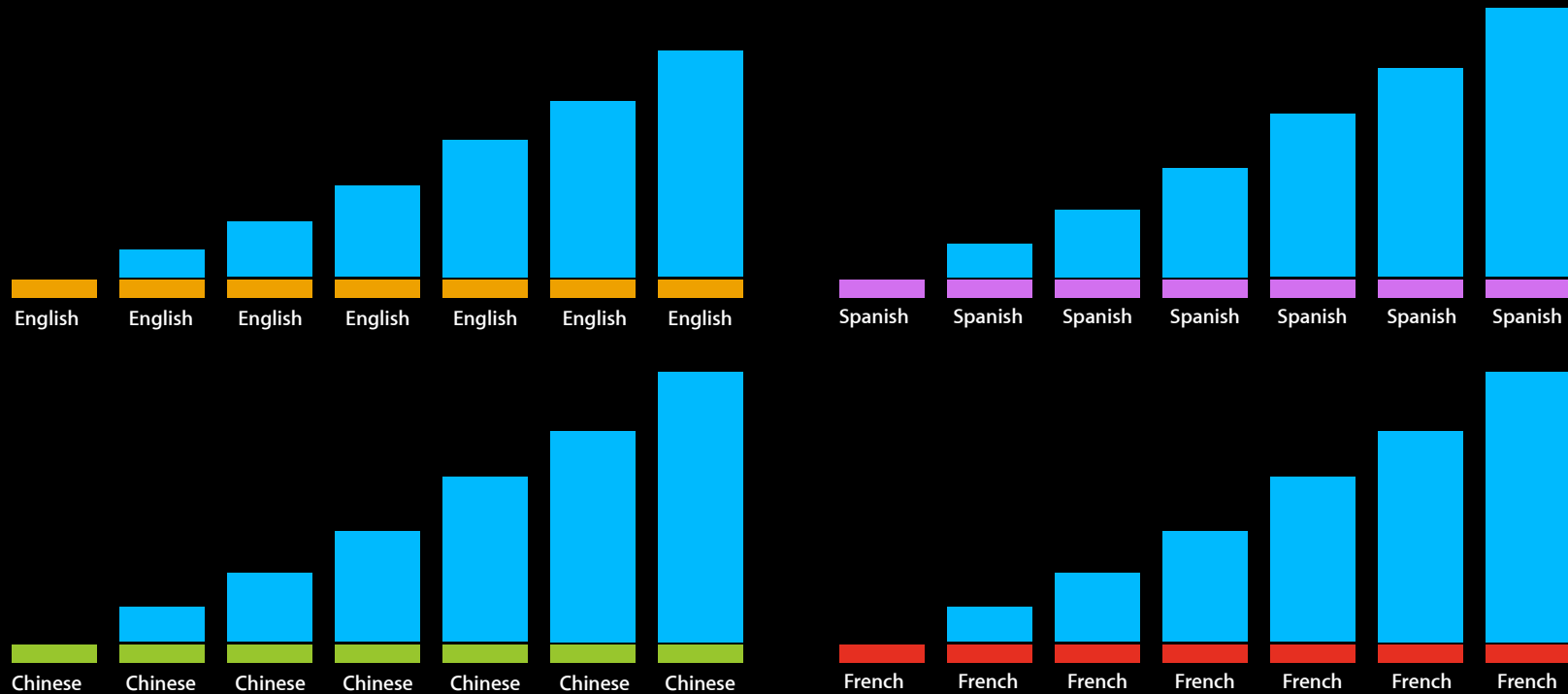
1 Language: 1.626 GB per Hour
4 Languages: 6.504 GB per Hour

Alternative Audio in iOS 5.0

- Audio can be packaged separately from video
 - allows multiple renditions of audio content
 - minimal storage overhead
- Can change audio while playing

Duplicating Video

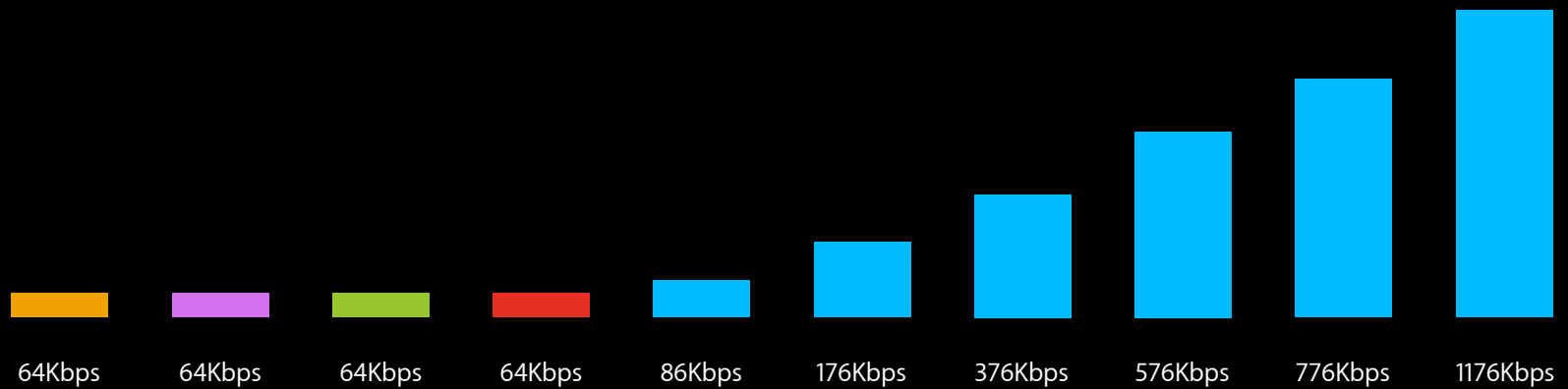
1 copy of video with 4 audio translations



1 Language: 1.626 GB per Hour
4 Languages: 6.504 GB per Hour

Alternative Audio in iOS 5.0

Audio packaged separately



Alternate audio allows us to reuse video and audio streams

1 Language: 1.378 GB per Hour
4 Languages: 1.462 GB per Hour

Specifying Alternative Audio

- Audio options are described in Variant Playlist:

```
#EXT-X-MEDIA:TYPE=AUDIO, GROUP-ID="aac", NAME="English",  
URI="english.m3u8", DEFAULT=YES
```

```
#EXT-X-MEDIA:TYPE=AUDIO, GROUP-ID="aac", NAME="Español",  
URI="spanish.m3u8"
```

```
#EXT-X-STREAM-INF:PROGRAM-ID=1, BANDWIDTH=190000, AUDIO="aac"  
lo/prog_index.m3u8
```

Specifying Alternative Audio

- Audio options are described in Variant Playlist:

```
#EXT-X-MEDIA:TYPE=AUDIO, GROUP-ID="aac", NAME="English",  
URI="english.m3u8", DEFAULT=YES
```

```
#EXT-X-MEDIA:TYPE=AUDIO, GROUP-ID="aac", NAME="Español",  
URI="spanish.m3u8"
```

```
#EXT-X-STREAM-INF:PROGRAM-ID=1, BANDWIDTH=190000, AUDIO="aac"  
lo/prog_index.m3u8
```

```
#EXT-X-STREAM-INF:PROGRAM-ID=1, BANDWIDTH=600000, AUDIO="aac"  
hi/prog_index.m3u8
```

Media Selection API

- Discover and select media options for playback

```
AVAsset *myAsset = getMyAsset();
NSArray *options = [myAsset
                   availableMediaCharacteristicsWithMediaSelectionOptions];

selectionGroup = [myAsset mediaSelectionGroupForMediaCharacteristic:
                  AVMediaCharacteristicAudible];

option = [[selectionGroup options] objectAtIndex:0];
AVPlayerItem *playerItem = [AVPlayerItem playerItemWithAsset: myAsset];
[playerItem selectMediaOption:option
               inMediaSelectionGroup:selectionGroup];
```

- Can switch dynamically

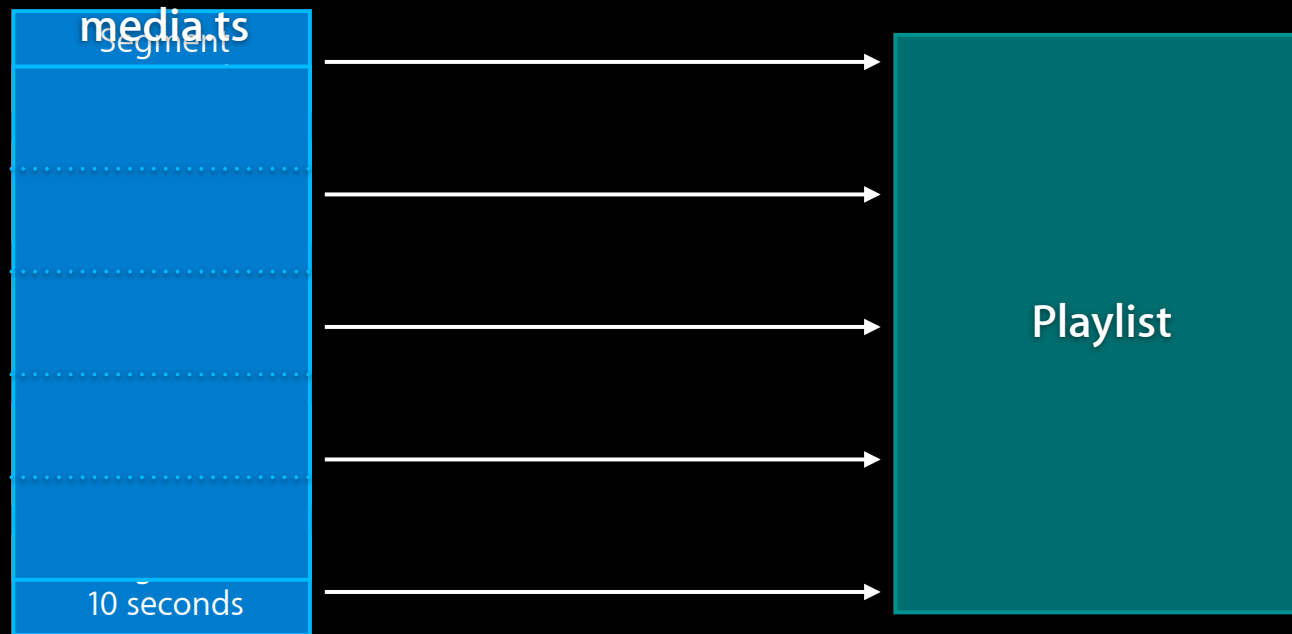
Demo

Bill May
Core Media Engineer

Byte-range Support

- Segment can be specified as a byte range of a URL
 - Allows segments to be consolidated into larger files
 - Improves cache associativity for CDN and proxy servers
 - Fewer files to manage

Byte-range Segments



Byte-range Support

- Segment can be specified as a byte range of a URL
 - Allows segments to be grouped into larger files
 - Improves cache associativity for CDN and proxy servers
 - Fewer files to manage

Byte-range Support

- Segment can be specified as a byte range of a URL
 - Allows segments to be grouped into larger files
 - Improves cache associativity for CDN and proxy servers
 - Fewer files to manage
- Media files should be static

Byte-range Support

- Segment can be specified as a byte range of a URL
 - Allows segments to be grouped into larger files
 - Improves cache associativity for CDN and proxy servers
 - Fewer files to manage
- Media files should be static
- New tag: `#EXT-X-BYTERANGE: length[@offset]`

Byte-range Sample Playlist

```
#EXTM3U
#EXTINF:10,
segment0.ts
#EXTINF:10,
segment1.ts
#EXTINF:10,
segment2.ts
```

```
#EXTM3U
#EXTINF:10,
#EXT-X-BYTERANGE:75232@0
media.ts
#EXTINF:10,
#EXT-X-BYTERANGE:82112@752321
media.ts
#EXTINF:10,
#EXT-X-BYTERANGE:69864@1573449
media.ts
```

Fast Forward and Reverse Playback

- Goal
 - Rapid forward and reverse playback (16x, 64x, -32x...)
 - For live content, too

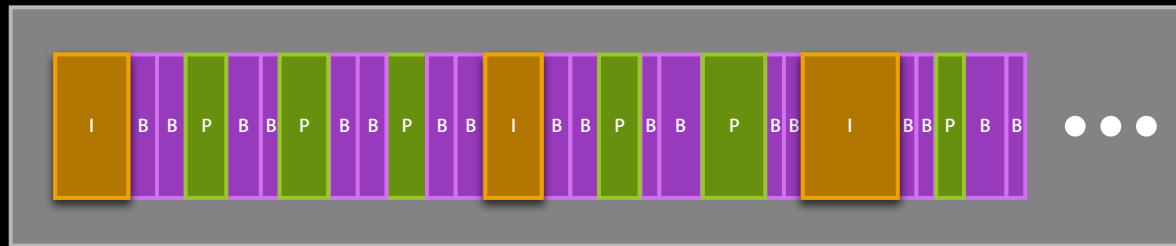
Fast Forward and Reverse Playback

- Goal
 - Rapid forward and reverse playback (16x, 64x, -32x...)
 - For live content, too
- Problem
 - Media segments cannot be downloaded quickly enough

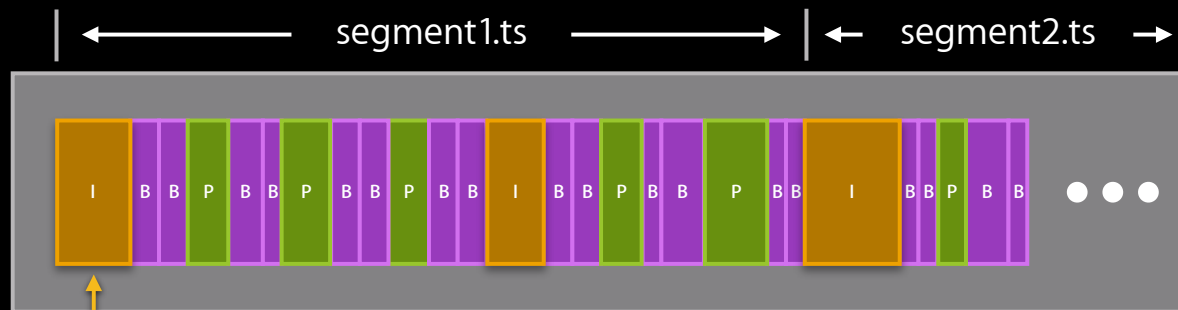
Fast Forward and Reverse Playback

- Goal
 - Rapid forward and reverse playback (16x, 64x, -32x...)
 - For live content, too
- Problem
 - Media segments cannot be downloaded quickly enough
- Solution
 - I-Frame-only playback

I-Frame-only Playback



I-Frame Playlist



```
#EXTM3U
#EXT-X-I-FRAMES-ONLY
...
#EXTINF:4.12,
#EXT-X-BYTERANGE:9400@376
segment1.ts
#EXTINF:3.56,
#EXT-X-BYTERANGE:7144@47000
segment1.ts
#EXTINF:3.82,
#EXT-X-BYTERANGE:10340@1880
segment2.ts
```

Provide Multiple I-Frame Playlists

```
#EXT-X-STREAM-INF:PROGRAM-ID=1,BANDWIDTH=1250000
```

```
BTBW0.8Mb/low.m3u8
```

```
#EXT-X-STREAM-INF:PROGRAM-ID=1,BANDWIDTH=3120000
```

```
BTBW2Mb/high.m3u8
```

Provide Multiple I-Frame Playlists

```
#EXT-X-STREAM-INF:PROGRAM-ID=1,BANDWIDTH=1250000
```

```
BTBW0.8Mb/low.m3u8
```

```
#EXT-X-STREAM-INF:PROGRAM-ID=1,BANDWIDTH=3120000
```

```
BTBW2Mb/high.m3u8
```

```
#EXT-X-I-FRAME-STREAM-INF:PROGRAM-ID=1,BANDWIDTH=380000
```

```
BTBW0.8Mb/iframes.m3u8
```

```
#EXT-X-I-FRAME-STREAM-INF:PROGRAM-ID=1,BANDWIDTH=657000
```

```
BTBW2Mb/iframes.m3u8
```

- More flexibility for good frame rate + high quality

Fast Forward and Reverse APIs

```
AVPlayerItem *myPlayerItem = GetMyPlayerItem();
AVPlayer *myPlayer = [AVPlayer playerWithPlayerItem: myPlayerItem];

if ( myPlayerItem.canPlayFastForward )
    myPlayer.rate = 32.0;

if ( myPlayerItem.canPlayFastReverse )
    myPlayer.rate = -20.0;
```

Demo

Bill May
Core Media Engineer

Closed Captions

- Accessibility text embedded in video stream
 - CEA-608
 - ATSC CC (A/53, Part 4)
- Can be displayed in iOS 4.2 and onward

Runtime Playback Statistics

- Access Log
 - Currently recorded for each period spent playing a variant
 - Start date / time
 - Duration downloaded, watched
 - URL
 - Server IP address (and # of changes)
 - Bandwidth information
 - Stall count
 - Dropped frame count
 - Playback GUID
- Error Log

Statistics API

- AVFoundation

```
AVPlayerItemAccessLog *accessLog = [myAVPlayerItem accessLog];  
AVPlayerItemErrorLog *errors = [myAVPlayerItem errorLog];  
NSData *bytes = [accessLog extendedLogData];
```

- MPMoviePlayerController

```
MPMovieAccessLog *accessLog = [myMoviePlayerController accessLog];  
MPMovieErrorLog *errors = [myMoviePlayerController errorLog];  
NSData *bytes = [accessLog extendedLogData];
```

Statistics—Formatted Log File

- Uses W3C Extended Log File Format

```
#Version: 1.0
```

```
#Software: AppleCoreMedia/1.0.0.9A220 (iPad; U; CPU OS 5_0 like Mac OS X;)
```

```
#Date: 2011/05/27 12:59:58.058
```

```
#Fields: date time uri s-ip c-duration-watched ...
```

```
2011/05/27 12:39:47.047 http://example.com/1500.m3u8 10.0.0.27 16.5 ...
```

```
2011/05/27 12:40:04.004 http://example.com/2000.m3u8 10.0.0.27 30.3 ...
```

```
2011/05/27 12:40:34.034 http://example.com/650.m3u8 10.0.0.27 27.0 ...
```

Best Practices

Delivery of Encrypted Media

- Distribute encrypted media files over regular HTTP

Delivery of Encrypted Media

- Distribute encrypted media files over regular HTTP
- Keys are loaded from URLs in the Playlist files

Delivery of Encrypted Media

- Distribute encrypted media files over regular HTTP
- Keys are loaded from URLs in the Playlist files
- Deliver keys over the network using HTTPS
 - Establish and require HTTPS session cookie, or
 - Put keys inside a protected HTTP Realm
 - Client must cache credentials before playback

Delivery of Encrypted Media

- Distribute encrypted media files over regular HTTP
- Keys are loaded from URLs in the Playlist files
- Deliver keys over the network using HTTPS
 - Establish and require HTTPS session cookie, or
 - Put keys inside a protected HTTP Realm
 - Client must cache credentials before playback
- Provide the keys directly using Application-defined URLs

Application-Defined URLs

- Define your own URL scheme
 - “abcd://foobar”

Application-Defined URLs

- Define your own URL scheme
 - “abcd://foobar”
- Implement NSURLProtocol
 - [NSURLProtocol registerClass:]
 - Called with URL—returns resource data or HTTP redirect
 - Not accessible from other Applications
- More secure and efficient than a localhost HTTP server

Application-Defined URLs

- Define your own URL scheme
 - “abcd://foobar”
- Implement NSURLProtocol
 - [NSURLProtocol registerClass:]
 - Called with URL—returns resource data or HTTP redirect
 - Not accessible from other Applications
- More secure and efficient than a localhost HTTP server
- Can be used for Keys and Playlists
 - Not media files

AirPlay of Encrypted Video Streams

- New in iOS 5.0
- Should just work if your app follows best practices
- Apps built for iOS 5.0 are opted in to AirPlay by default
 - Opt out if you need to

Chaining Players Does Not Work Well

- Sometimes used to present pre-roll and mid-roll ads
- Each Player requires separate buffering stage
- Players compete for bandwidth, unpredictably
- Will not retain optimal bit rate
- Difficult to implement seek
- Very difficult to implement FF/RW

Use Single Playlist for Program + Ads

- Put ads directly into main Playlist
 - DISCONTINUITY tags between ads and program content
 - Use floating-point EXTINF durations
 - `-[AVPlayer addBoundaryTimeObserverForTimes:usingBlock:]`
- Compose Playlists on server or from within App
 - Application-defined URLs (NSURLProtocol)

Serve Playlists with gzip Encoding

- Automatic, on demand
- ~10x compression for long m3u8 files

- For apache:

```
<IfModule deflate_module>  
    AddOutputFilterByType DEFLATE application/vnd.apple.mpegurl  
    Header append Vary Accept-Encoding env=!dont-vary  
</IfModule>
```

Accurate Content Description

- CODECS attribute

`#EXT-X-STREAM-INF:BANDWIDTH=190000,CODECS="avc1.42e00a,mp4a.40.2"`

- Can differentiate audio-only streams from audio/video

Accurate Content Description

- CODECS attribute

`#EXT-X-STREAM-INF: BANDWIDTH=190000, CODECS="avc1.42e00a, mp4a.40.2"`

- Can differentiate audio-only streams from audio/video

- RESOLUTION attribute

`#EXT-X-STREAM-INF: BANDWIDTH=560000, RESOLUTION=640x480`

Accurate Content Description

- CODECS attribute

`#EXT-X-STREAM-INF: BANDWIDTH=190000, CODECS="avc1.42e00a, mp4a.40.2"`

- Can differentiate audio-only streams from audio/video

- RESOLUTION attribute

`#EXT-X-STREAM-INF: BANDWIDTH=560000, RESOLUTION=640x480`

- Floating-point EXTINF durations

`#EXTINF: 9.57,`

Accurate Content Description

- CODECS attribute

`#EXT-X-STREAM-INF: BANDWIDTH=190000, CODECS="avc1.42e00a, mp4a.40.2"`

- Can differentiate audio-only streams from audio/video

- RESOLUTION attribute

`#EXT-X-STREAM-INF: BANDWIDTH=560000, RESOLUTION=640x480`

- Floating-point EXTINF durations

`#EXTINF: 9.57,`

- Pay attention to the mediastreamvalidator

TLS Version 1.2

- iOS 5.0 connects with TLS 1.2
 - Formerly known as SSL, used for HTTPS
 - Previous iOS releases used TLS 1.0
 - Many new cipher suites are presented in ClientHello

TLS Version 1.2

- iOS 5.0 connects with TLS 1.2
 - Formerly known as SSL, used for HTTPS
 - Previous iOS releases used TLS 1.0
 - Many new cipher suites are presented in ClientHello
- Testing with some apps has revealed incompatibilities

TLS Version 1.2

- iOS 5.0 connects with TLS 1.2
 - Formerly known as SSL, used for HTTPS
 - Previous iOS releases used TLS 1.0
 - Many new cipher suites are presented in ClientHello
- Testing with some apps has revealed incompatibilities
- Test your apps

Selecting TARGETDURATION

- Controls Playlist reload interval
- Indicates maximum segment duration
 - Impacts maximum IDR interval
- Does not affect startup (buffering) time

Content Production

- Do not restart encoding on segment boundaries
 - Format, track count, CC, GOP structure, etc. must be continuous
 - Unless starting a DISCONTINUITY

Content Production

- Do not restart encoding on segment boundaries
 - Format, track count, CC, GOP structure, etc. must be continuous
 - Unless starting a DISCONTINUITY
- Start each TS segment with PAT/PMT
 - Each segment requires only one PAT/PMT

Content Production

- Do not restart encoding on segment boundaries
 - Format, track count, CC, GOP structure, etc. must be continuous
 - Unless starting a DISCONTINUITY
- Start each TS segment with PAT/PMT
 - Each segment requires only one PAT/PMT
- Other formats have coarser sample time scales (e.g. 1 ms)
 - Timestamp accuracy has been lost due to rounding
 - Must restore this accuracy when transcoding to MPEG2 90,000

To Summarize...

- HTTP Live Streaming continues to evolve
- Stay current
- Bring your questions to the Graphics and Media lab
- Give us your feedback

Related Sessions

Exploring AV Foundation

Presidio
Tuesday 2:00PM

AirPlay and External Displays in iOS apps

Presidio
Tuesday 3:15PM

Working with Media in AV Foundation

Pacific Heights
Wednesday 2:00PM

Introducing AV Foundation Capture For Lion

Pacific Heights
Wednesday 3:15PM

Capturing from the Camera using AV Foundation on iOS 5

Pacific Heights
Wednesday 4:30PM

Labs

AirPlay Lab	Graphics, Media & Games Lab B Wednesday 9:00AM
AV Foundation Lab	Graphics, Media & Games Lab C Wednesday 9:00AM
HTTP Live Streaming Lab	Graphics, Media & Games Lab D Wednesday 9:00AM
QT Kit Lab	Graphics, Media & Games Lab A Wednesday 9:00AM
AV Foundation Lab	Graphics, Media & Games Lab B Thursday 9:00AM
QuickTime Lab	Graphics, Media & Games Lab D Thursday 9:00AM
DAL Lab	Graphics, Media & Games Lab C Thursday 9:00AM

More Information

Documentation

HTTP Live Streaming Resources

<http://developer.apple.com/resources/http-streaming/>

Timed metadata

[http://developer.apple.com/library/ios/#documentation/AudioVideo/Conceptual/](http://developer.apple.com/library/ios/#documentation/AudioVideo/Conceptual/HTTP_Live_Streaming_Metadata_Spec/)

[HTTP_Live_Streaming_Metadata_Spec/](http://developer.apple.com/library/ios/#documentation/AudioVideo/Conceptual/HTTP_Live_Streaming_Metadata_Spec/)

Apple Developer Forums

<http://devforums.apple.com>

Eryk Vershen

Media Technologies Evangelist

evershen@apple.com

