Encoding for Adaptive Streaming

Jan Ozer
www.streaminglearningcenter.com
jozer@mindspring.com/
276-238-9135

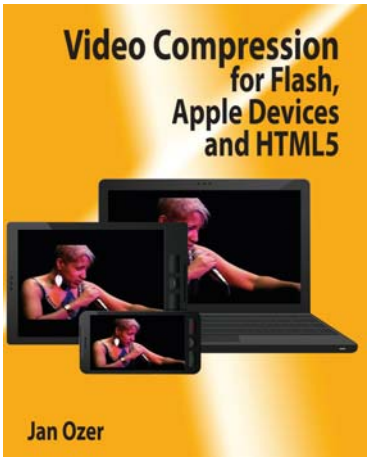
Encoding for Adaptive Streaming

- Introduction to adaptive streaming
- How they work
  - RTMP (server-based)
  - HTTP (no server)
- Generic issues
  - How many streams/what data rates
- Encoding for adaptive streaming
- Transmuxing

For Additional Reference

The material in this presentation was derived from Jan Ozer's book, *Video Compression for Flash, Apple Devices and HTML5.*

For further explanation of the concepts presented in this presentation, check out the book at bit.ly/ozerbook1



Technology Overview

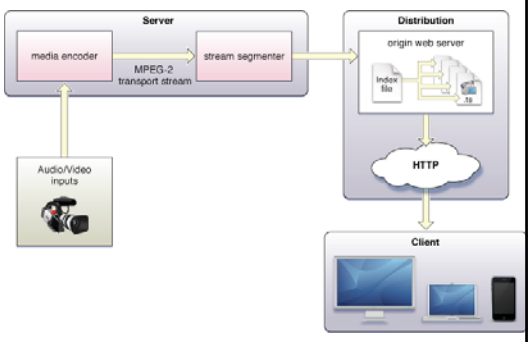
- Streams switched to adapt to factors like:
  - Changing delivery bandwidths (avoid hard stops)
  - CPU utilization at client (avoid frame drops)
- Information is gathered by player
  - Server-based systems (RTMP Flash) *deliver* a different stream when change is required, switching at key frame
  - HTTP-based systems (HTTP Flash, iOS) use 2-10 second file chunks
    - Player *retrieves* chunk from different source file to effectuate stream switch (more later)

RTMP Flash - Overview

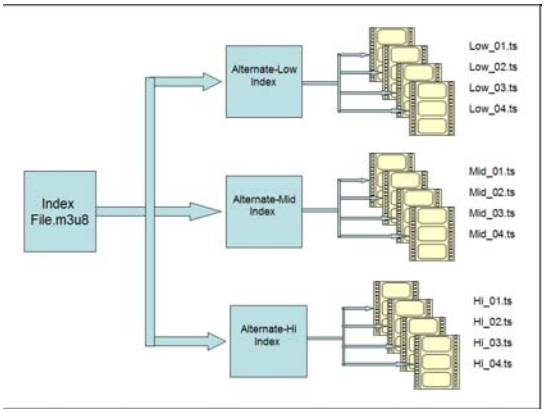
- Server driven system
  - Uses files as encoded (no chunking)
    - MP4, F4V, FLV, whatever
  - Server switches as necessary at key frame

Apple HTTP Live Streaming: How it works

- Encoding
  - Encode as normal, send to segmenter
  - Files chunked, inserted into transport stream (.ts extension)
  - Manifest file (M3U8) created
  - Uploaded to server
- Client
  - Monitors heuristics
  - Changes retrieved stream as necessary

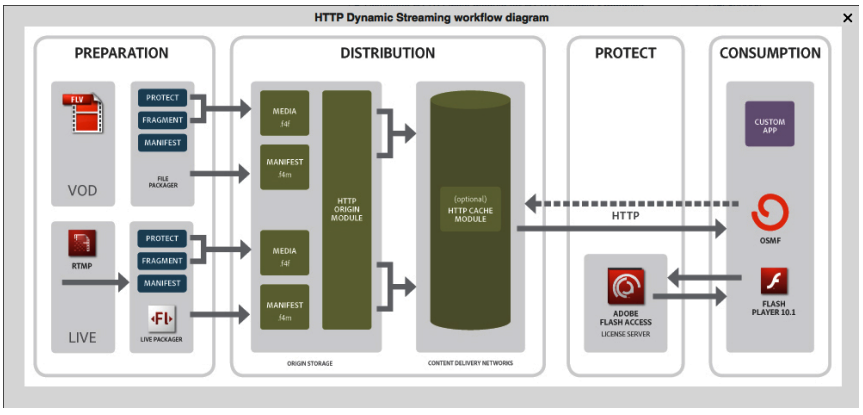


The Manifest File Structure



Streaming Learning Center

HTTP Flash - How it Works



Streaming Learning Center

To Boil it Down-HTTP Flash

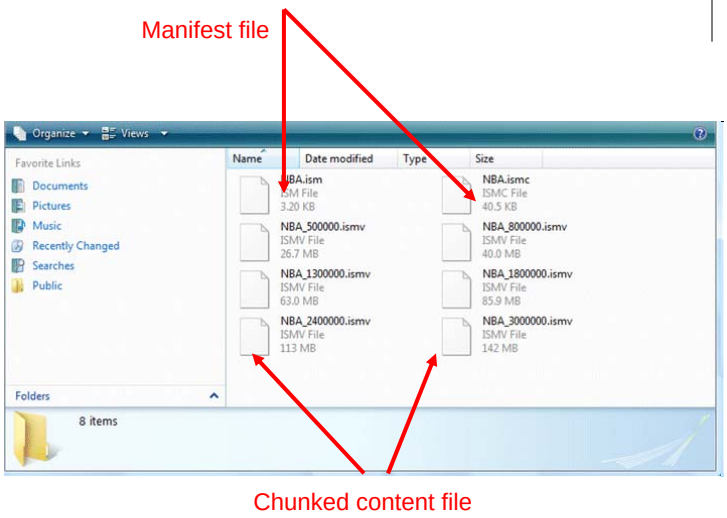
MP4 → Into Packager

- .F4F — Media file chunks
- .F4M — Manifest files that tell player where to find file chunks

- Player driven system
  - Player monitors heuristics
  - Retrieves different stream as necessary

Streaming Learning Center

Smooth Streaming – How It Works



Streaming Learning Center

Summary

- RTMP
  - Server driven
  - No special encoding requirement
- HTTP
  - No server
  - Files must be chunked; metadata created
  - Player does all the work

Generic Encoding Issues

- How many streams
  - What configuration
- What data rates

Inputs for Analysis

- Multiple case studies (Harvard, Turner, MTV, NBC, Indiana University)
  - *StreamingMedia* here: bit.ly/ozeradaptive
- White papers and other guides as identified throughout
- Several consulting projects

How Many Streams?

- Considerations
  - Enough to provide relevant coverage
    - SD – 3-4 (320x240 – 640x480)
    - HD – need additional streams to provide full spectrum
  - MTV
    - At least one stream for each window size served

How Many Streams?

- Considerations
  - Number of window sizes served

Scenario	Format	Frame Size	Total Bitrate	Audio Bitrate	bits/pixel *frame @ 30 fps	bits/pixel *frame @ 24 fps
Mobile & constrained (low)	baseline, mono, 10 fps	448x252	150	48	0.09	0.09
Mobile & constrained (high)	baseline, mono	448x252	450	48	0.12	0.15
Sidebar placements	main profile, stereo	384x216	400	96	0.12	0.15
Small in-page	main profile, stereo	512x288	750	96	0.15	0.18
Medium in-page	main profile, stereo	640x360	1200	96	0.16	0.20
Large in-page	main profile, stereo	768x432	1700	96	0.16	0.20
Full size in-page	main profile, stereo	960x540	2200	96	0.14	0.17
HD 720p (full screen)	high profile, stereo	1280x720	3500	96	0.12	0.15

MTV.com

How Many Streams?

- MTV Schema
  - Have at least one for each window size served (optimum quality when stream size=window size)
  - Never switch to stream larger than current viewing size
    - So, if playing in 640x360 window, don't switch to 768x432 unless viewer expands window or goes full screen

How Many Streams?

- Other considerations
  - Subscription – if paid, more streams for better service
    - Major League Baseball – 11 streams
  - More if entertainment
    - MTV with 8, though Turner (PGA golf) with 4
  - Fewer if education or corporate
    - Indiana - 3, Harvard, 5

Stream Count – Bottom Line

- There are no hard and fast rules
  - More streams required for HD than SD
    - Just a function of size
  - More for entertainment than education/business
    - Entertainment – about the experience
    - Business – it's about making sure the viewer can watch the stream
  - More for subscription than general entertainment
    - Provide more options when viewer is paying

What Resolution?

- Other considerations
  - Never encode at larger than source
    - Scaling upwards degrades quality
  - Very few producers adhere to mod-16 (where height/width are divisible by 16)
    - But all favor at least mod-8, all at least mod-4
- SD adaptive - many producers use the same resolution for all streams (640x480)

A Word About Frame Rates

- At extremely low data rates, often more effective to drop frame rate than resolution
  - So:
    - 640x480@ 15 fps is better than
    - 320x240@ 30 fps
- In particular, Apple recommends dropping frame rate on lowest data rate streams

Format Specific: HTTP Live Streaming

16:9 Aspect Ratio	Dimensions	Total Bit Rate	Video Bit Rate	Keyframe	Restrict Profile to:	WORKS ON iPod Touch Gens 2, 3, 4 iPhone 3G 3GS	WORKS ON iPhone 4 iPad 1, 2
CELL	480x320	64	na	na	na	*	*
CELL	416x234	150	110	30	Baseline, 3.0	*	*
CELL	416x234	240	200	45	Baseline, 3.0	*	*
CELL	416x234	440	400	90	Baseline, 3.0	*	*
WIFI	640x360	640	600	90	Baseline, 3.0	*	*
WIFI	640x360	1240	1200	90	Main, 3.1		*
WIFI	960x540	1840	1800	90	Main, 3.1		*
WIFI	1280x720	2540	2500	90	Main, 3.1		*
WIFI	1280x720	4540	4500	90	Main, 3.1		*

Bit.ly/bestpracticehttplive

Format Specific: HTTP Live Streaming

- Adobe white paper
  - Cluster around window sizes
    - Switches least apparent w/in same window size

16x9				
Stream #	Picture Size	V	A	AV
1	256x144	150	64	214
2	256x144	250	64	314
3	512x288	450	64	514
4	512x288	600	64	664
5	512x288	800	64	864
6	512x288	1200	64	1264
7	768x432	1400	64	1464
8	1280x720	1700	64	1764
9	1280x720	2500	64	2564
10	1280x720	3500	64	3564
11	1920x1080	4200	64	4264
12	1920x1080	5300	64	5364

bit.ly/Levkovhttp

Format Specific: Smooth Streaming

	Olympics	Sunday Night Football	Expression Encoder	Recommend
How many streams	6 per input, 8 total configurations	7 for HD input, 6 for SD input	10 for 1080p, 8 for all others	6-8
Resolution	Most are mod-16, all at least mod-8	2 are mod-4, all others mod-16	Most mod-16, all at least mod-4	At least mod-4, try for mod-16 in smaller sizes
Data rate (kbps)	400, 600, 900, 950, 1250, 1600, 1950, 3450	50, 350, 600, 950, 1500, 2250, 3450	230, 331, 477, 688, 991, 1400, 2100, 3000, 5000, 6000	Smooth Streaming Calculator
Frame rate	29.97 for all streams	29.97 except lowest quality stream (15 fps)	All match source	Source
Profile/Level	NA	NA	Main	Main
Entropy Encoding	NA	NA	CABAC	CABAC
B-frame interval	1	1	1	1
Reference frames	Unknown	Unknown	4	4
VBR/CBR	CBR for live, 110% constrained VBR - on demand	All live, all CBR	CBR - 1 pass	CBR
Key frame interval/chunk size	2 seconds	2 seconds	2 seconds	2 seconds
Client side buffer	5 seconds	5 seconds	4 seconds	4 seconds
Audio parameters	48 kbps for all streams	48 kbps for all streams	160kbps stereo audio for all streams	48 kbps mono for all streams

Streaming Learning Center

Format Specific: RTMP/HTTP Flash

- Adobe Media Encoder Presets

Preset	Resolution	Video bitrate (kbps)	Profile	Key frame	Audio bitrate (kbps)	Channels
Phone and Tablet, 3G, 16x9	512x288	300	Baseline	72	48	Stereo
Phone and Tablet, 3G, 16x9	512x288	450	Baseline	72	48	Stereo
Phone and Tablet, 3G, 16x9	512x288	650	Baseline	72	48	Stereo
PC &TV, SD, Med, 16x9	768x432	1,140	Main	72	64	Stereo
PC &TV, SD, High 16x9	768x432	1,740	Main	72	64	Stereo
PC &TV, HD, Med, 16x9	1280x720	2,440	High	72	64	Stereo
PC &TV, SD, High 16x9	1280x720	3,440	High	72	64	Stereo

http://bit.ly/adaptivepresets

Streaming Learning Center

What Data Rates?

- Considerations
  - Must be sufficiently far apart to avoid constant switching
  - Will vary with stream size
    - Smaller gaps at lower bandwidths
    - Larger gaps at higher bandwidths

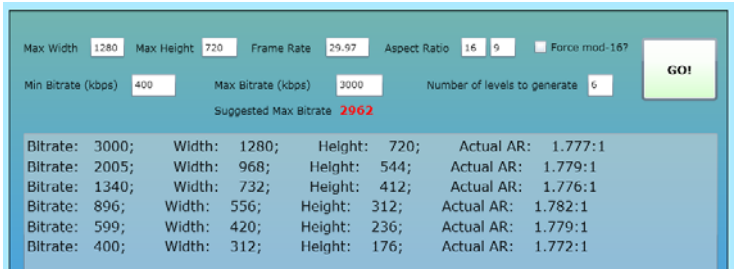
What Data Rates?

- MTV Schema

Scenario	Format	Frame Size	Total Bitrate	
Mobile & constrained (low)	baseline, mono, 10 fps	448x252	150	
Mobile & constrained (high)	baseline, mono	448x252	450	
Sidebar placements	main profile, stereo	384x216	400	350 kbps
Small in-page	main profile, stereo	512x288	750	450 kbps
Medium in-page	main profile, stereo	640x360	1200	500 kbps
Large in-page	main profile, stereo	768x432	1700	500 kbps
Full size in-page	main profile, stereo	960x540	2200	1300 kbps
HD 720p (full screen)	high profile, stereo	1280x720	3500	

What Data Rates?

- http://alexzambelli.com/WMV/MBRCalc.html
  - For Smooth Streaming, but good starting point
  - Enter parameters, calculates rez/data rate for you
  - Attempts to keep quality consistent at all data rates



How should H.264 encoding parameters change?

- Now we know stream count, resolution and data rate
- How do we customize encoding for adaptive?
  - Key frame settings
  - Bit rate control
  - Audio parameters

Key Frame Interval

- Why important - RTMP Flash – stream switches occur at key frame
  - Key frame location must be identical in all streams
    - Use same interval
    - Disable scene change detection
  - Typically shorter (2-5 seconds) to enable more responsive switching

Streaming Learning Center

Key Frame Interval

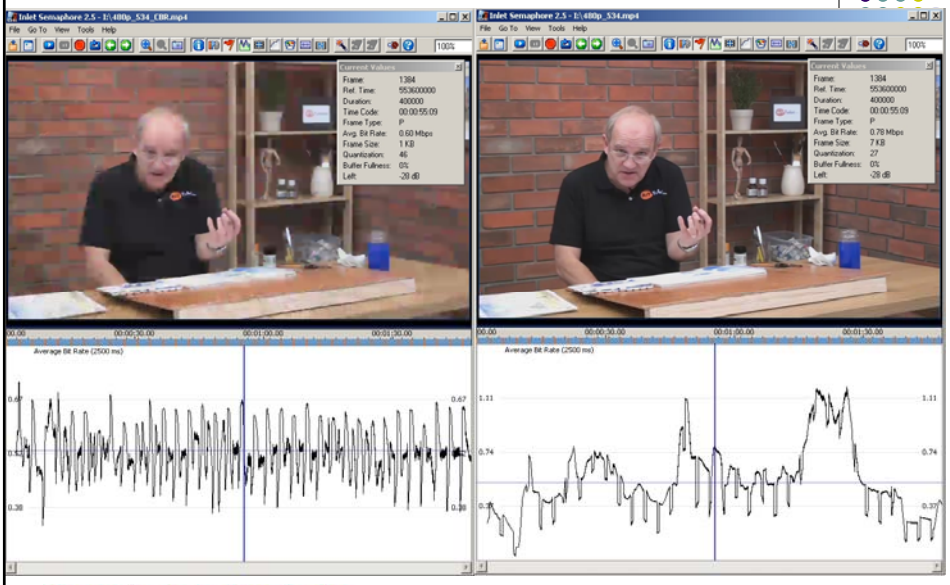
- Why important – HTTP technologies - key frame must be first frame of every chunk
  - Key frame interval must divide evenly into chunk duration
    - If 9-sec. chunks, use key frame interval of 3 sec
- As with RTMP, key frame location must be identical in all streams, so:
- Use same interval
  - Disable scene change detection

Streaming Learning Center

In General - VBR vs CBR

- CBR more conservative
  - Produces smoothest stream, which is easiest to consistently deliver
- But, quality is typically inferior to VBR
  - Sometimes noticeably so

Streaming Learning Center

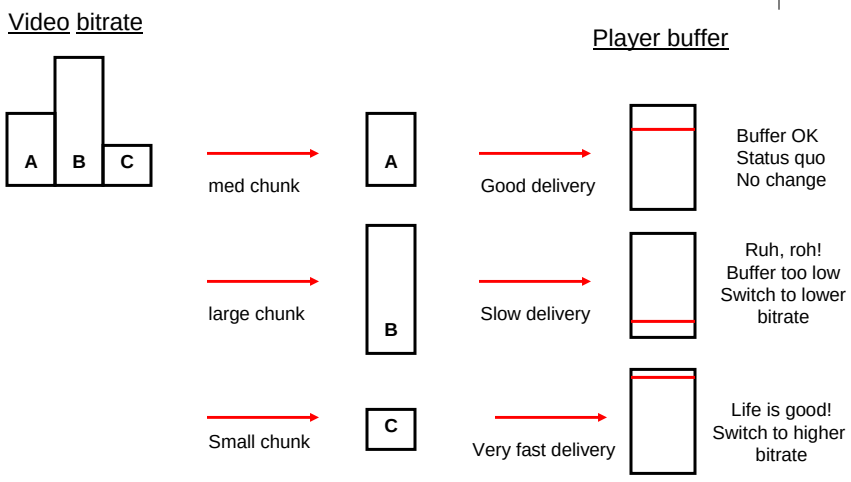


Streaming Learning Center

In General - VBR vs CBR

- VBR
  - Variable data rate
  - Could introduce stream switches that relate to bit rate control rather than change in conditions
  - Too many stream switches degrade perceived quality

VBR vs CBR



What's the Answer?

- Most conservative
  - Use CBR
- Many real world producers
  - Use constrained VBR,
  - Less constrained at high data rates, where difference in bandwidths is great (MTV uses 2X constrained)

In General - Audio Parameters

- Most conservative - use same parameters for all files
  - Popping can occur if audio parameters change
  - But, doesn't optimize experience at higher bit rates
- If you do switch audio parameters
  - Switch from stereo to mono at same per channel sampling rate and bit rate
    - From 128 kbps/44 kHz/16-bit/stereo to 64 kbps/44 kHz/16-bit/mono
  - Test to ensure no artifacts when switching streams

H.264 Encoding Parameters

- Adapt to target playback device
  - Adobe white paper/encoder
    - AME - Baseline for lowest data rate streams, on up to High
    - White paper – Main to High
- Adapt to lowest common denominator
  - If streaming to Apple devices
    - Baseline – iPhone/iPod touch – through 4G
    - Main – iPhone 4/iPod touch 4G/iPad

WebM-based Adaptive Streaming

- Nascent market;
  - Some technology providers
    - Anevia
    - Quavlive
  - HTML5 standard coming in a DASH
    - Dynamic Adaptive Streaming over HTTP (DASH)
    - http://mpeg.chiariglione.org/working_documents.htm
  - No encoding recommendations at this point

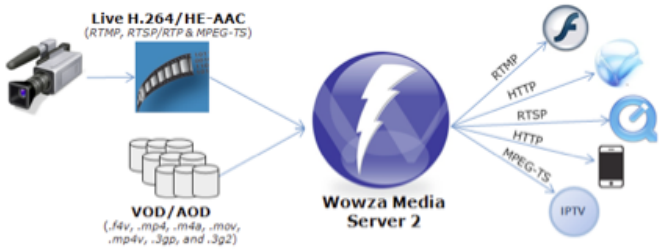
Transmuxing Technologies

- Most producers must serve two clients
  - Flash (or Silverlight) AND
  - iOS (and now Android)
- In the past, that meant two separate encoding and delivery workflows
- Now, multiple technologies for:
  - "Transmuxing" H.264 stream
  - Using correct protocol to distribute to target

Transmuxing

- Not re-encoding
  - "re-wrapping" file into a different container format
  - Creating necessary manifest files
  - Delivering files to proper server
  - All in real time

Transmuxing Technologies



- Options
  - Technology providers - Wowza, Microsoft, Adobe
  - Service providers - Akamai (in the network repackaging)
- Key point:
  - If serving multiple targets, you must produce using lowest common denominator H.264 encoding parameters

New Feature

- Wowza Transcoder
  - Single H.264 file in
  - Re-encode into different adaptive streams
  - Transmux as necessary
  - Shipped in the last couple of weeks
- Sorenson Squeeze Live

RTMP Flash Sources

- Abhinav Kapoor, “Live dynamic streaming with Flash Media Server 3.5,” (adobe.ly/kapoorlivefms)
- Maxim Levkov, “Video encoding and transcoding recommendations for HTTP Dynamic Streaming on the Flash Platform,” (adobe.ly/Levkovhttp)
- Larry Bouthillier, “How to do Dynamic Streaming with Flash Media Server,” website (bit.ly/fmsdynamic)
- “Tutorial: On-demand HTTP Dynamic Streaming,” Adobe website (bit.ly/ondemanddynamic)
- “Encoding Guidelines Dynamic Streaming for Flash over HTTP,” Akamai website (bit.ly/akamaiwhitepaper)

HTTP Flash Resources

- Best sources:
  - Maxim Levkov - Adobe - very detailed recommendations
    - adobe.ly/Levkovhttp
  - Akamai - White paper:”Encoding Guidelines for Dynamic Streaming for Flash over HTTP”
    - bit.ly/akamaiwhitepaper

HLS Resources

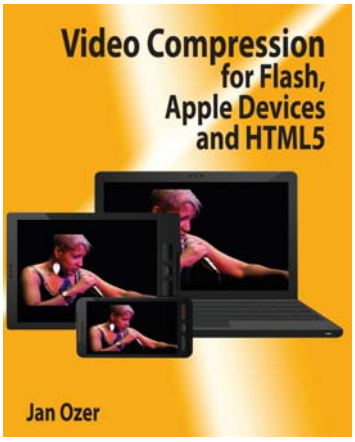
- Sources
  - Apple Tech Note: “Best Practices for Creating and Deploying HTTP Live Streaming Media for the iPhone and iPad,” (bit.ly/bestpracticehttplive)
  - Apple Tech Note: “HTTP Live Streaming Overview,” (bit.ly/httpliveoverview)
  - Apple Tech Note: “Using HTTP Live Streaming,“ (bit.ly/usinghttplive)

Silverlight Resources

- “Akamai HD for Microsoft Silverlight On-Demand Encoding Recommendations,” bit.ly/akamaiwhitepapers
- Alex Zambelli, “IIS Smooth Streaming Technical Overview,” bit.ly/smoothstreamzambelli
- Zambelli webcast – bit.ly/smtranscoding

Questions?

- For more information, check out the book
  - Available on Amazon
  - Some copies available today
    - US $29.95 retail
    - Show special: 10 pounds or Euros



Streaming Learning Center