

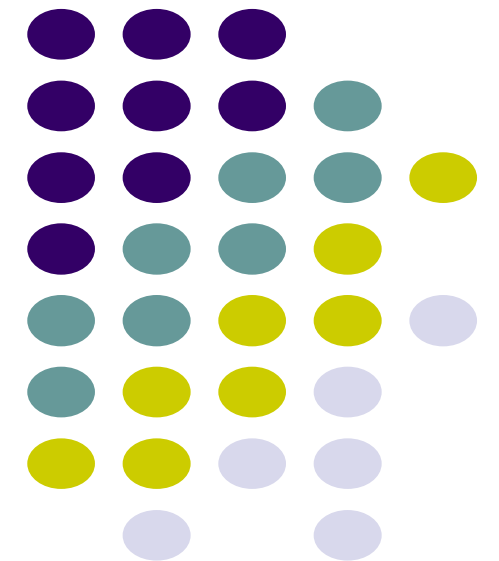
Encoding for iDevices

Jan Ozer

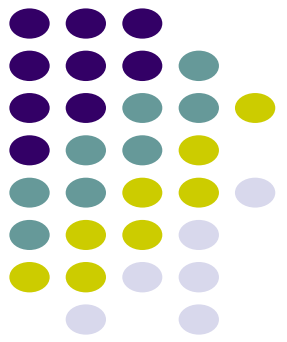
www.streaminglearningcenter.com

[jozer@mindspring.com/](mailto:jozer@mindspring.com)

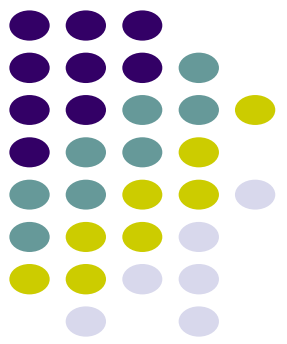
[276-238-9135](tel:276-238-9135)



Agenda



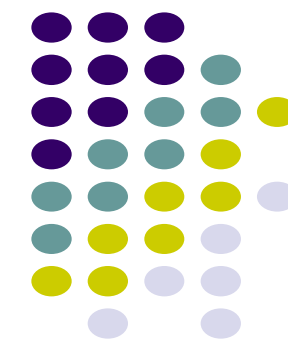
- Producing H.264 for iTunes
 - What is H.264
 - How to configure for iDevices
- Encoding for the iPad/iPhone
 - Tethered delivery via iTunes
 - HTTP Live Streaming



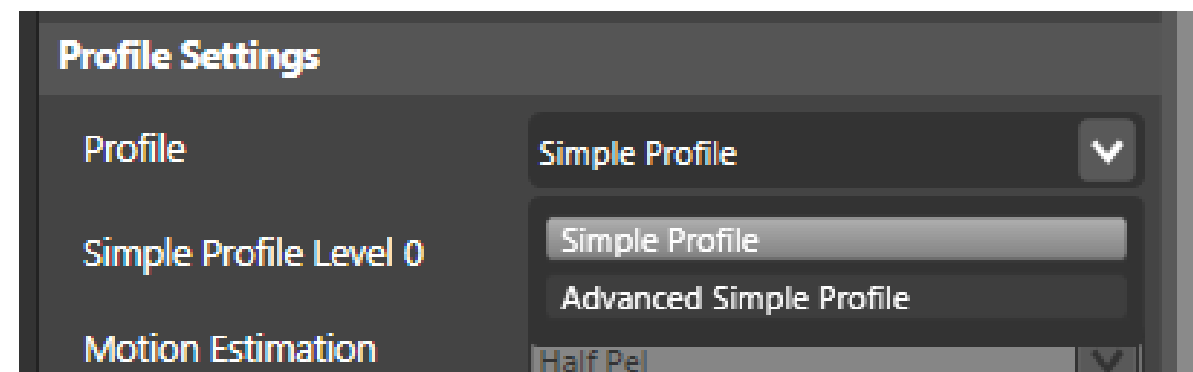
MPEG-4 Specification

- Introduced in 1998 by ISO/IEC Moving Picture Experts Group
- 28 “Parts” within specification
 - Part 2 - MPEG video codec (the MPEG-4 codec)
 - Part 3 - MPEG-4 audio (AAC, etc)
 - Part 10 - Advanced Video Coding (AVC/H.264)
 - Part 14 - container format (MP4)

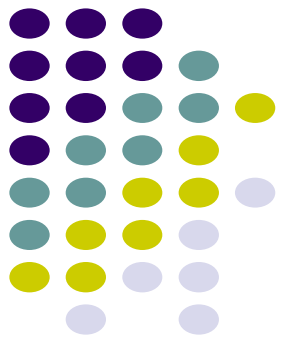
The MPEG-4 Video Codec



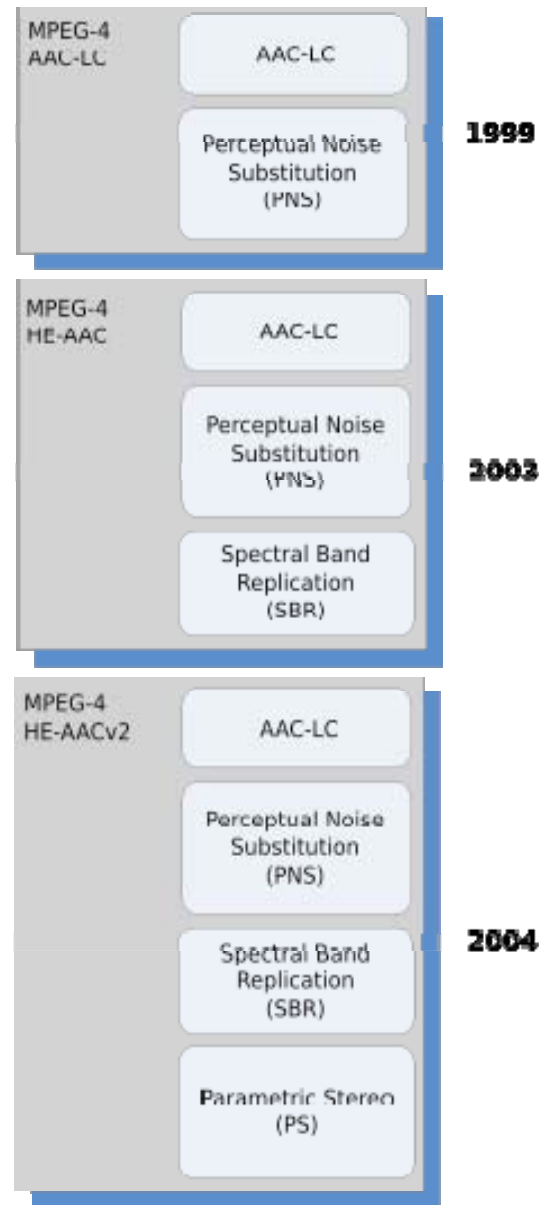
- Used only for low power devices:
Two profiles:
 - Simple Profile - very low power, low bandwidth applications
 - Advanced Simple Profile - Simple plus:
 - Support for "MPEG"-style quantization
 - Support for B pictures (a.k.a. B-frames)
 - Motion compensation



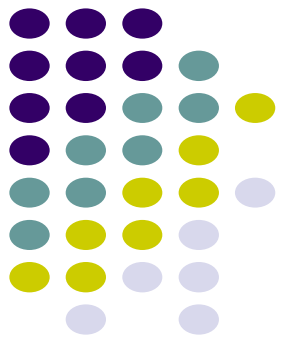
- Check specs on target devices
 - Will refer to MPEG-4 encoding only in mobile segments
 - Never use for computer playback



MPEG-4 Audio



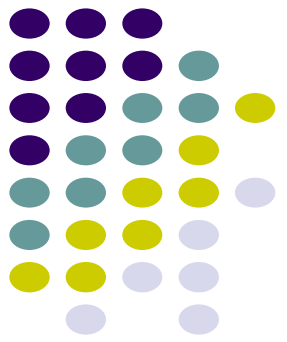
- AAC-Low Complexity (AAC-LC)
 - The most basic and most broadly compatible
- High Efficiency AAC (2003)
 - Also called AAC+, aacPlus and
- High Efficiency AACv2 (2006)
 - Also called enhanced aacPlus, Enhanced AAC+, aacPlus v2 and eAAC+



What are H.264 Profiles?

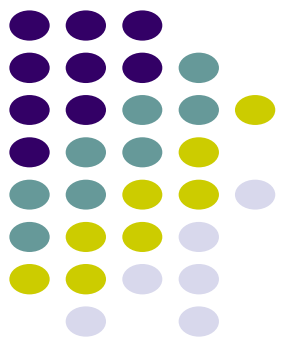
- “Define a set of coding tools or algorithms that can be used in generating a bitstream”

	Baseline	Extended	Main	High
I and P Slices	Yes	Yes	Yes	Yes
B Slices	No	Yes	Yes	Yes
Multiple Reference Frames	Yes	Yes	Yes	Yes
In-Loop Deblocking Filter	Yes	Yes	Yes	Yes
CAVLC Entropy Coding	Yes	Yes	Yes	Yes
CABAC Entropy Coding	No	No	Yes	Yes
Interlaced Coding (PicAFF, MBAFF)	No	Yes	Yes	Yes
8x8 vs. 4x4 Transform Adaptivity	No	No	No	Yes
Quantization Scaling Matrices	No	No	No	Yes
Separate Cb and Cr QP control	No	No	No	Yes
Separate Color Plane Coding	No	No	No	No
Predictive Lossless Coding	No	No	No	No
	Baseline	Extended	Main	High



Which Profile?

- Critical to know your target profile before encoding
 - Device
 - iPod/iPhone - always Baseline
 - iPad - Main
 - Computer playback - High for all targets
- Issues to consider
 - iPad/iPhone/iPod Touch – one file for all, use Baseline
 - Computer/iPad - use Main

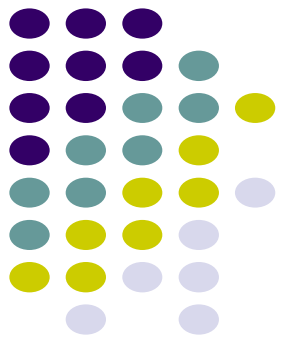


What are H.264 Levels?

- “Constrains key parameters in the bitstream”

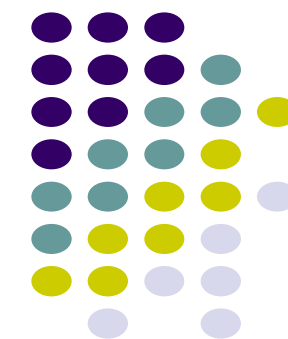
Level number	Max video bit rate (VCL) for Baseline, Extended and Main Profiles	Max video bit rate (VCL) for High Profile	Examples for high resolution @ frame rate (max stored frames) in Level
1	64 kbit/s	80 kbit/s	128x96@30.9 (8) 176x144@15.0 (4)
1b	128 kbit/s	160 kbit/s	128x96@30.9 (8) 176x144@15.0 (4)
1.1	192 kbit/s	240 kbit/s	176x144@30.3 (9) 320x240@10.0 (3) 352x288@7.5 (2)
1.2	384 kbit/s	480 kbit/s	320x240@20.0 (7) 352x288@15.2 (6)

Production for iDevices

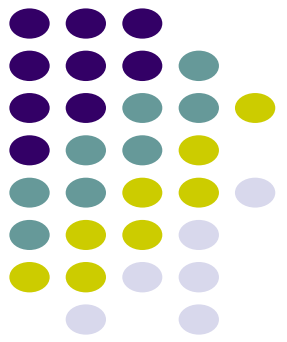


- Bottom line:
 - Choose encoding tool that you trust
 - Use preset as adjusted herein
 - Don't tinker

Producing for iDevices



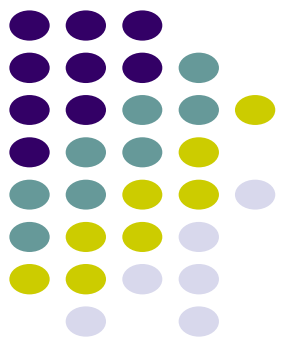
- Two scenarios
 - Video podcasts
 - Covered next
 - Streaming to iDevices
 - Best done with HTTP Live Streaming, covered later



iDevice Specification Overview

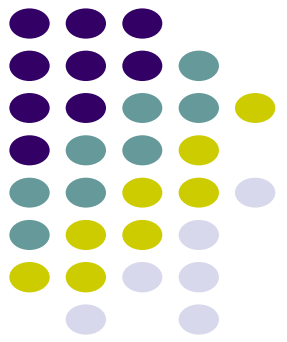
	Original iPod (pre-5g)	iPod Nano/Classic	iPod touch/ iPhone	iPhone 4/iPod touch	iPad 1&2
Device spec					
Screen rez	320x240	320x240	480x320	960x640	1024x768
Aspect ratio	4:3	4:3	16:9-ish	16:9-ish	4:3
Codec spec					
Video codec	H.264	H.264	H.264	H.264	H.264
Max data rate	768 kbps	2.5 Mbps	2.5 Mbps	14 Mbps	14 Mbps
Max video rez	320x240	640x480	640x480	720p	720p
Frame rate	30 fps	30 fps	30 fps	30 fps	30 fps
Profile/level	Baseline to Level 1.3	Baseline to Level 3.0	Baseline to Level 3.0	Main to Level 3.1	Main to Level 3.1
Audio codec	AAC-LC	AAC-LC	AAC-LC	AAC-LC	AAC-LC
Max data rate	160 kbps	160 kbps	160 kbps	320 kbps	320 kbps
Audio params	48 kHz, stereo	48 kHz, stereo	48 kHz, stereo	48 kHz, stereo	48 kHz, stereo
Container formats	m4v/mp4/mov	m4v/mp4/mov	m4v/mp4/mov	m4v/mp4/mov	m4v/mp4/mov

- Limited number of devices, very well defined
- 3 categories, low, medium and high



Tethered Deliver via iTunes

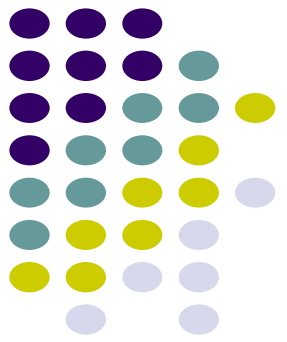
- Decisions, decisions
- Survey results
- Recommended encoding parameters
 - 320x240
 - 640x360
 - 720p



Decision Time

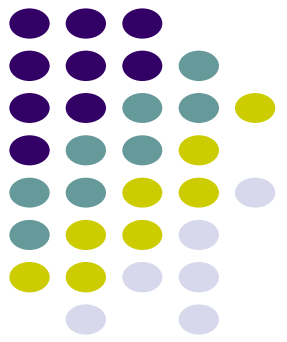
- Decision 1:
 - Abandon older iPods that supports only 320x240?
- Decision 2:
 - Support low and high resolution iDevices with single max 640x480 stream?
- Decision 3:
 - Distribute multiple files?

iTunes Survey



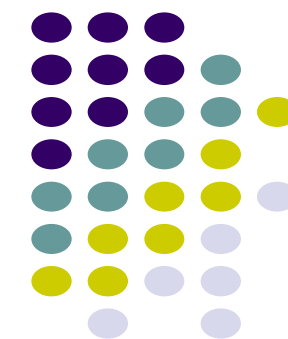
- 48 files from 34 different producers
 - Three letter networks
 - Prominent technology and other popular sites
 - Featured podcasts
 - All FREE downloads

Findings



- Abandon 320x240?
 - 9 of 48 produced at 320x240 or lower
 - 1 produced at 640x480 with MPEG-4 codec
 - Lower quality, but should play on older devices
- Go exclusively big screen?
 - 11 of 12 producing at 720p also produced at 640x480 or lower
 - If you go big, you should also go small
- Distribute multiple files?
 - 12 produced **same show** in different sizes
 - 4 others produced **different shows** in different sizes
 - 18 were single size only

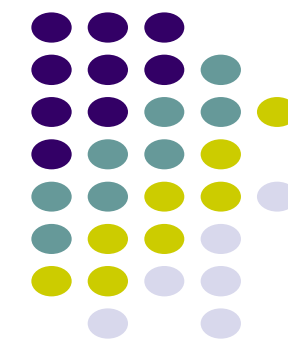
Music Videos



Music Videos	Width	Height	Total Pixels	Data Rate	FPS	Audio Data Rate	Bits per Pixel	Profile
Eminem/Rihanna	640	256	163,840	1,457	23.976	256	0.371	Baseline
Ke\$sha	640	360	230,400	1,565	23.976	251	0.283	Baseline
Lady Gaga	640	464	296,960	1,514	23.976	249	0.213	Baseline
Michael Jackson - Thriller	640	464	296,960	1,512	29.97	256	0.170	Baseline
Shakira	640	352	225,280	1,533	25	251	0.272	Baseline
Taylor Swift	640	352	225,280	1,513	23.976	248	0.280	Baseline

- All 640x480 or smaller
 - Play on all but the oldest devices, not tuned for new ones
- All 256 kbps audio

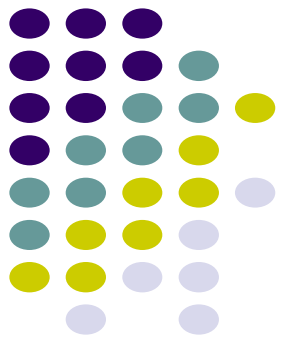
HD TV Episodes



HD TV Episodes	Width	Height	Total Pixels	Data Rate	FPS	Audio Data Rate	Bits per Pixel
House	1280	720	921,600	3,871	23.976	160	0.175
Lie to Me	1280	720	921,600	4,298	23.976	157	0.195
Nikita	1280	720	921,600	4,141	23.976	157	0.187
Back to School	1280	720	921,600	4,273	23.976	156	0.193
Law and Order	1280	720	921,600	4,116	23.976	155	0.186

- All 720P @ 23.98 fps
- All .175 to .193 bits per pixel
- All 160 kbps audio target

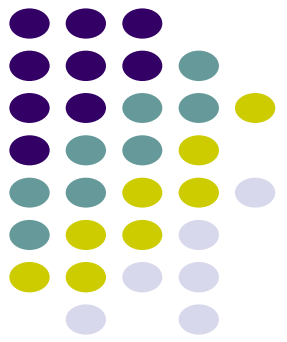
SD TV Episodes



SD TV Episodes	Width	Height	Total Pixels	Data Rate	FPS	Audio Data Rate	Bits per Pixel
House	640	480	307,200	1,235	23.976	123	0.168
Law and Order	640	480	307,200	1,459	23.976	124	0.198
Bad Girls Club	640	480	307,200	1,460	23.976	128	0.198
Nikita	640	480	307,200	1,520	23.976	125	0.206
Back to School	640	480	307,200	1,518	23.976	125	0.206
Real Housewives of B Hills	640	360	230,400	1,539	29.97	122	0.223
Real Housewives of DC	640	480	307,200	1,424	29.97	106	0.155

- All 640x480 or smaller
- All but oldest, not tuned for new ones
- All 256 kbps audio

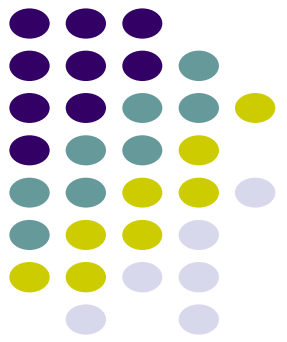
Encoding Parameters - 320x240



	320x240 - Episode	Survey
Video codec	H.264 codec, Baseline profile	H.264 codec, Baseline profile
Data rate average	600 kbps	528 kbps (average)
Frame rate	match source	match source
Audio	AAC Low	AAC Low
Data rate	128 kbps/stereo	111 kbps/stereo

- Episode preset conforms to what producers are doing.

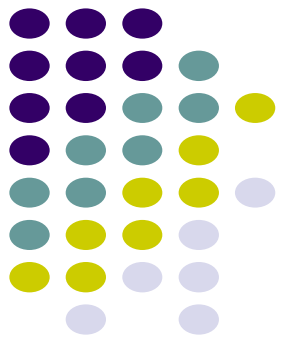
Encoding Parameters - 640x360



	640x480 - Episode	Survey
Video codec	H.264 codec, Baseline profile	H.264 codec, Baseline profile
Data rate average/max	1.2 mbps	1.319 mbps
Frame rate	match source	match source
Audio	AAC Low	AAC Low
Data rate	128 kbps/stereo	114 kbps/stereo

- Episode preset conforms – might be a bit low – be sure to test quality (Demo – adjust data rate)

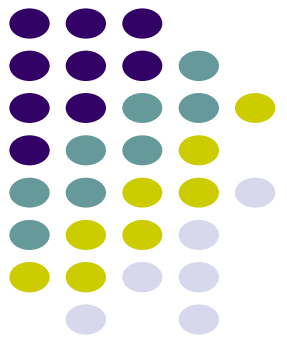
Encoding Parameters - 720p



	720p – Episode	Survey
Video codec	H.264 codec, Main profile	H.264 codec, 11 of 12 are Main or High profile
Data rate average/max	4.4 mbps	2.845 mbps
Frame rate	match source	match source
Audio	AAC Low	AAC Low
Data rate	128 kbps/stereo	134 kbps/stereo

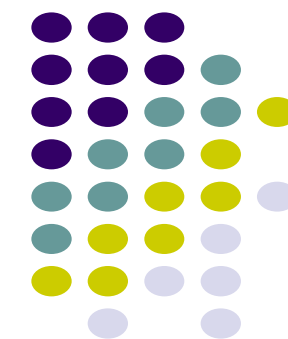
- Episode preset is a bit high for video; experiment with lower rates (demo – adjust downward)

Encoding for Adaptive Streaming



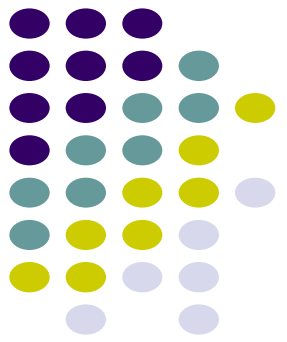
- Introduction
- Transmux strategies
- HTTP Live Streaming to iOS/Android

Adaptive Streaming - Introduction



- Concept
 - Customize experience for viewer device and bandwidth
 - High power/high bandwidth – great experience
 - Lower power/low bandwidth – lesser experience, but it plays
 - Adapt to changing conditions
 - All transparent to the viewer

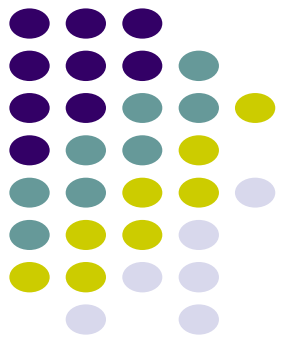
Major League Baseball



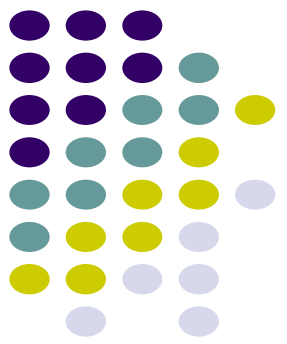
- Example
 - MLB offers 11 streams in subscription service
 - Intelligent player
 - Monitors CPU
 - Monitors buffer level
 - System adjusts speed to ensure optimal quality stream



Advantages of Adaptive Streaming



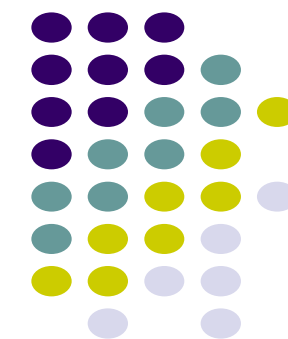
- Enables highest quality viewing experience
 - Can create very high quality streams because the system will shift to lower quality if required
 - Rewards high performance/high bitrate consumers while still serving those at the other end of the spectrum



Technology Overview

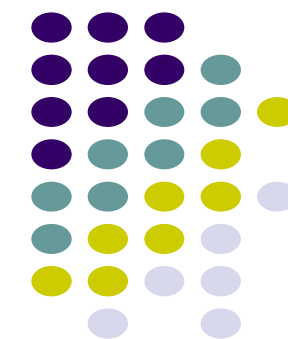
- Streams switched to adapt to factors like:
 - Changing delivery bandwidths (avoid hard stops)
 - CPU utilization at client (avoid frame drops)
- Information is gathered by player
 - Server-based systems (RTMP Flash) *deliver* a different stream when change is required, switching at key frame
 - HTTP-based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

Encoding for Adaptive Streaming



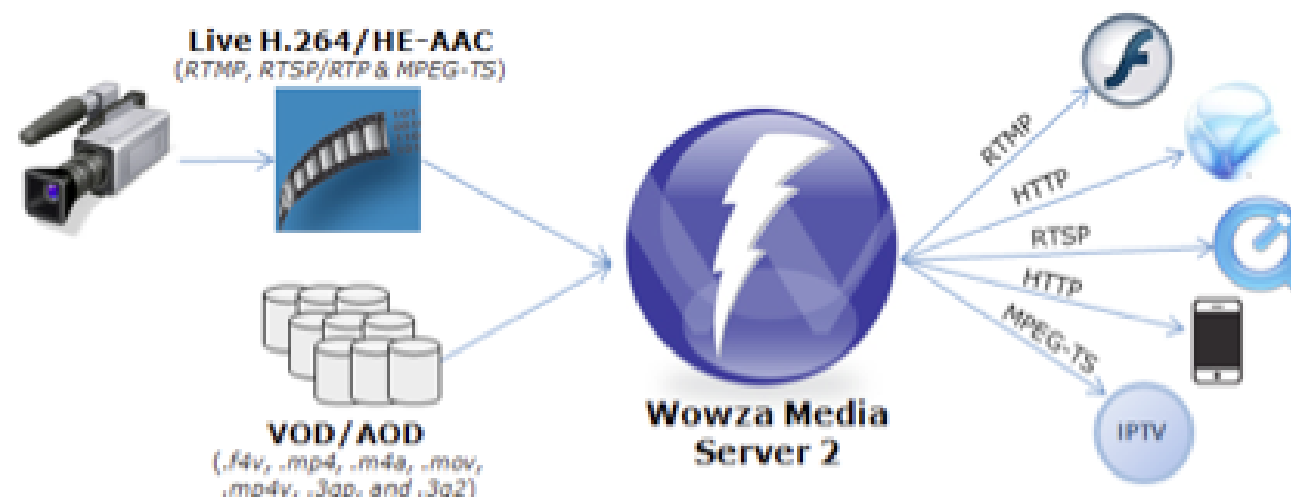
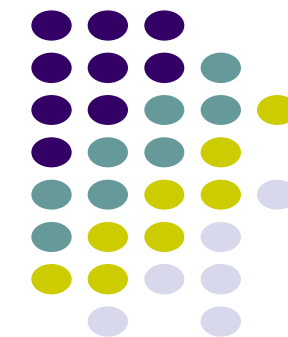
- How to configure streams to:
 - Optimize playback experience across all served devices and bandwidths
 - How many streams, what resolutions, what data rates
 - Work within requirements of adaptive streaming technology
 - Key frame interval, VBR vs. CBR, audio parameters

Transmuxing Technologies



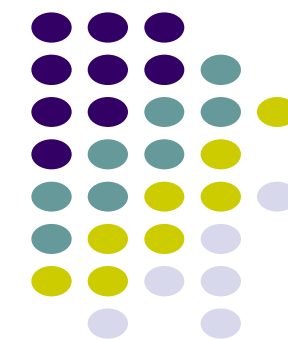
- Most producers must serve two clients
 - Flash (or Silverlight)
 - iOS (and now Android)
- In the past, that meant two separate encoding and delivery workflows
- Now, multiple technologies for:
 - “Transmuxing” H.264 stream
 - Using correct protocol to distribute to target

Transmuxing Technologies



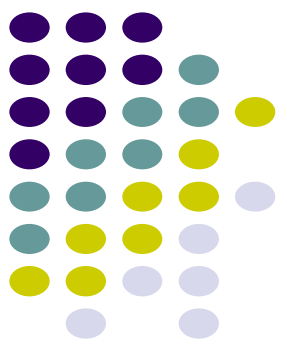
- Options
 - Technology providers - Wowza, Microsoft, Adobe
 - Service providers - Akamai (in the network repackaging)
- Key point:
 - If serving multiple targets, you must produce using lowest common denominator H.264 encoding parameters

Apple HTTP Live Streaming

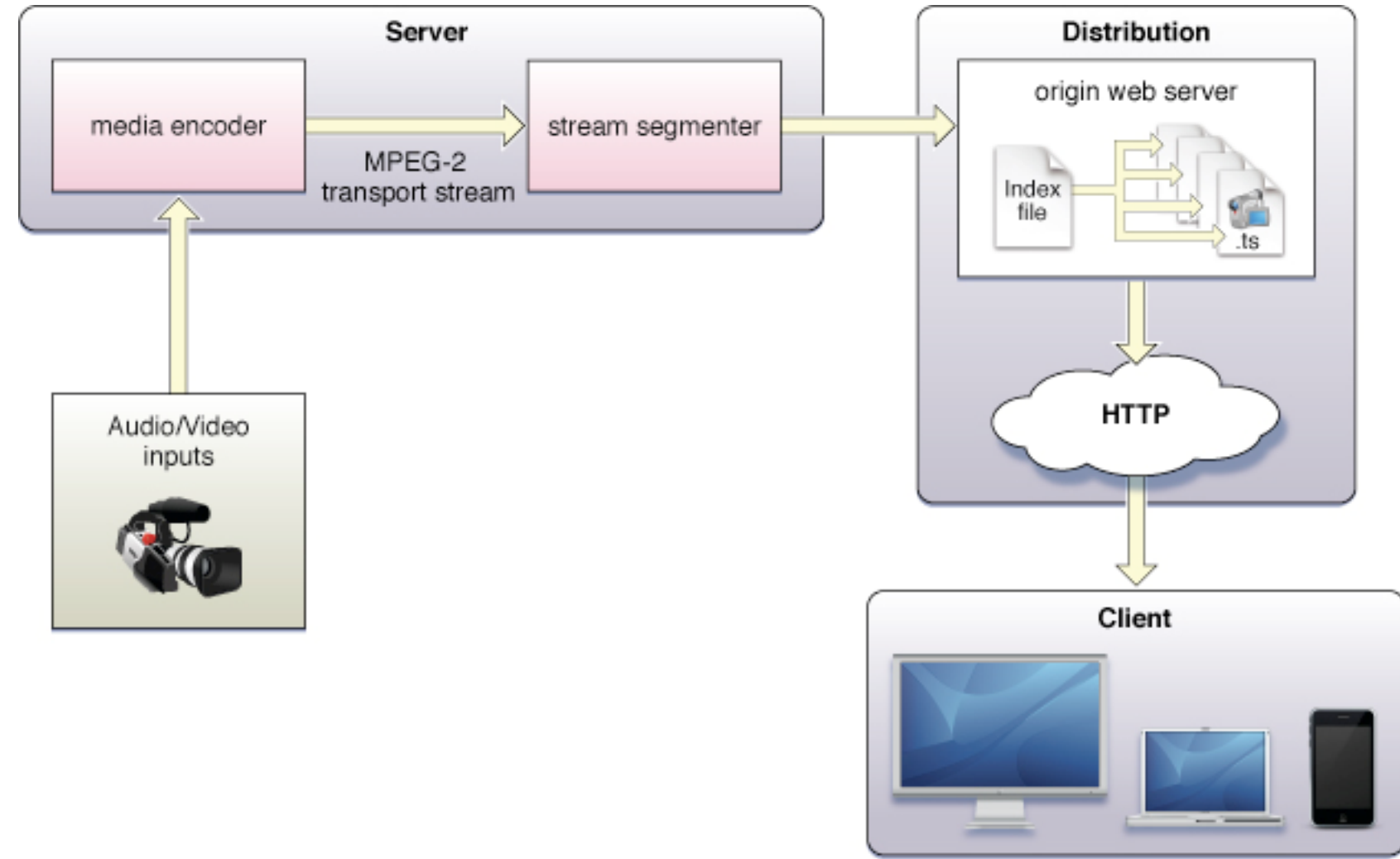


- Primary technology for iOS devices
 - Also supported in Android 3.0
- How it works
- How to customize your encoding

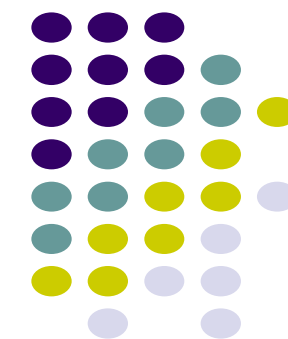
Apple HTTP Live Streaming: How it works



- 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 file as necessary



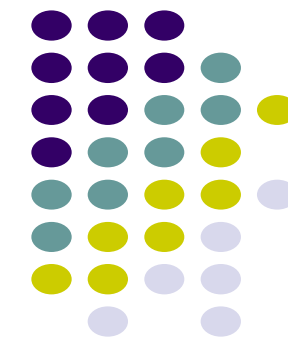
Encoding for HTTP Live Streaming



- 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)

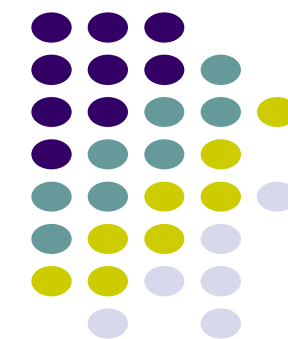
HTTP Live Streaming



iPad								
16:9 Aspect Ratio								
	Dimensions	Frame Rate *	Total Bit Rate	Video Bit Rate	Audio Bit Rate	Audio Sample Rate	Keyframe	Restrict Profile to:
CELL	480x320	na	64**	na	40	22.05	na	na
CELL	400x224	10	150	110	40	22.05	30	Baseline, 3.0
CELL	400x224	12 to 15	240	200	40	22.05	45	Baseline, 3.0
CELL	400x224	29.97	440	400	40	22.05	90	Baseline, 3.0
WIFI	640x360	29.97	640	600	40	22.05	90	Baseline, 3.0
WIFI	640x360	29.97	840	800	40	22.05	90	Main, 3.1
WIFI	640x360	29.97	1240	1200	40	22.05	90	Main, 3.1
4:3 Aspect Ratio								
	Dimensions	Frame Rate *	Total Bit Rate	Video Bit Rate	Audio Bit Rate	Audio Sample Rate	Keyframe	Restrict Profile to:
CELL	480x320	na	64**	na	40	22.05	na	na
CELL	400x300	10	150	110	40	22.05	30	Baseline, 3.0
CELL	400x300	12 to 15	240	200	40	22.05	45	Baseline, 3.0
CELL	400x300	29.97	440	400	40	22.05	90	Baseline, 3.0
WIFI	640x480	29.97	640	600	40	22.05	90	Baseline, 3.0
WIFI	640x480	29.97	840	800	40	22.05	90	Main, 3.1
WIFI	640x480	29.97	1240	1200	40	22.05	90	Main, 3.1

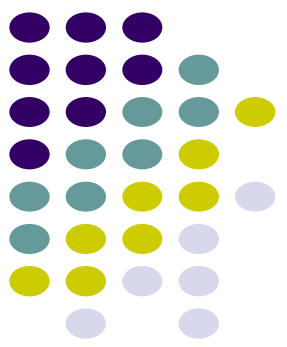
[Bit.ly/bestpracticehttplive](http://bit.ly/bestpracticehttplive)

HTTP Live - Encoding Parameters

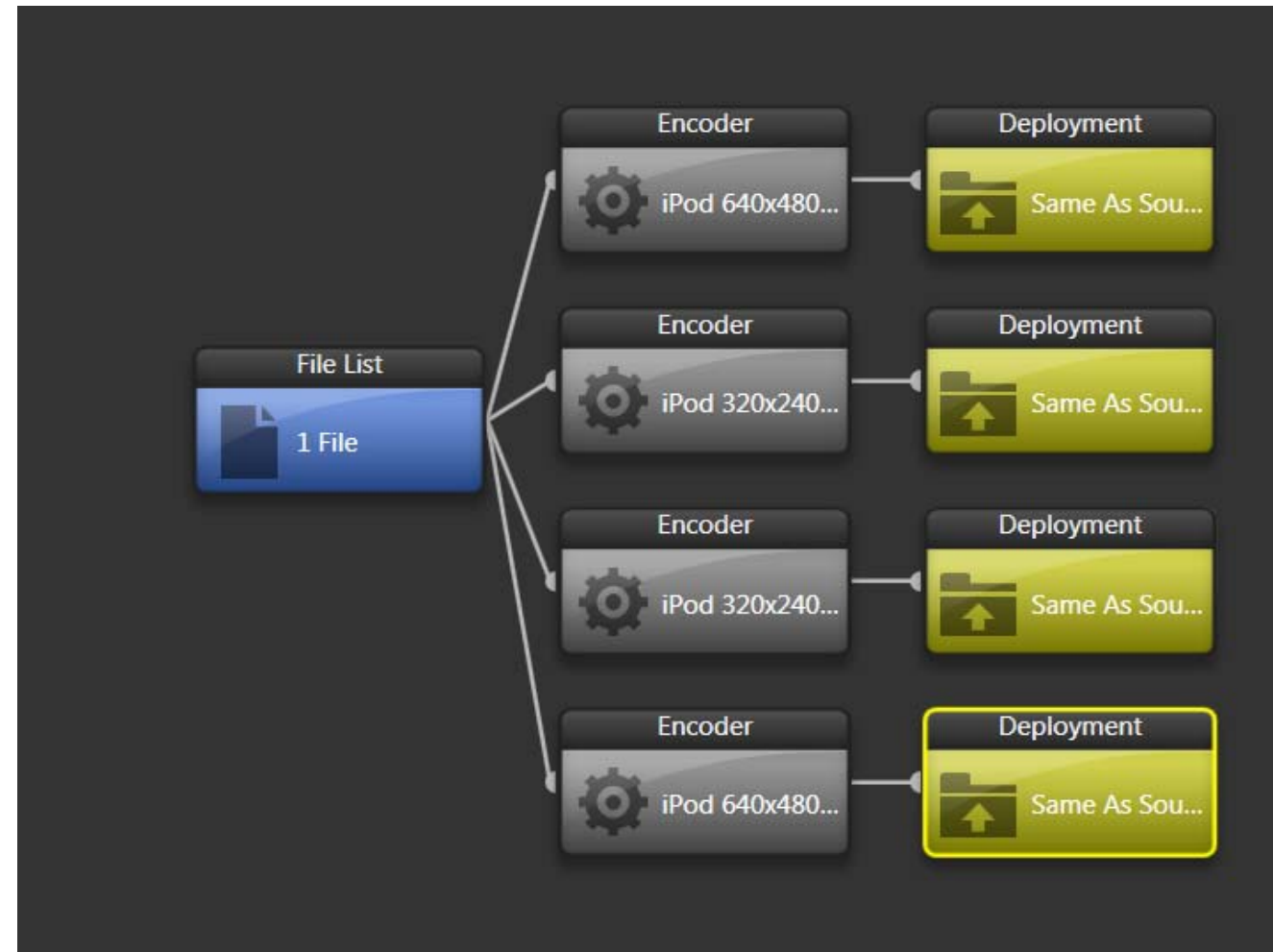


- Apple gives very good guidance
- Filling in the blanks
 - Profile/level - optimize for target
 - M3U8 can be device sensitive - won't send older iPods to iPad/iPhone 4 stream
 - VBR/CBR
 - As discussed
 - Key frame - chunked technology, optimal if key frame divides evenly into chunk duration

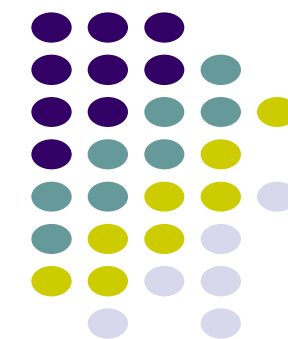
Encoding Options for HTTP Live Streaming



- Episode
 - One workflow with multiple output

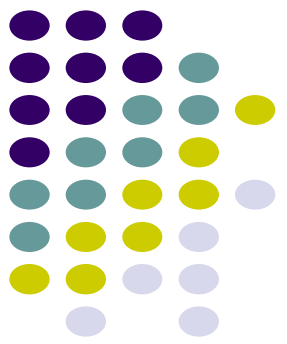


Static File Delivery



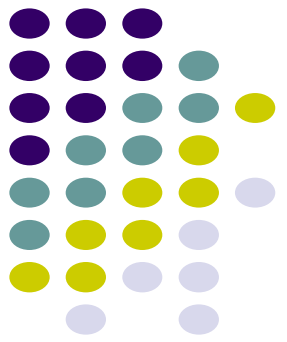
- Adaptive via HTTP Live Streaming is preferred delivery technique
- When not using adaptive, you can offer:
 - High quality stream that only viewers on fast connections can view
 - Lowest common denominator stream that plays everywhere but doesn't look so great
 - Multiple streams, selectable by the viewer

Recommended Encoding Parameters – Static Delivery



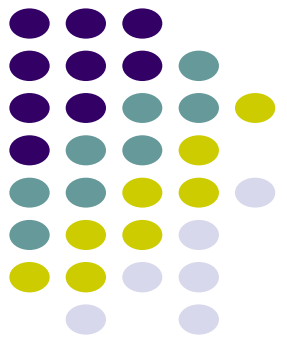
Encoding Parameters	iPad-Only Stream	iDevice - Mid Quality Stream	iDevice - Low Quality Stream
Video			
Resolution	640x360	400x224	400x224
Frame rate	Full frame rate	Full frame rate	1/3 frame rate
Profile/Level	Baseline/3.1	Baseline/Level 3	Baseline/Level 3
Bitrate control	CBR	CBR	CBR
Video data rate	600 kbps	400 kbps	110 kbps
Key frame interval	3 seconds (90 frames)	3 seconds (90 frames)	3 seconds (90 frames)
Audio	AAC-LC, 40 kbps, mono, CBR	AAC-LC, 40 kbps, mono, CBR	AAC-LC 40 kbps, mono, CBR

Demo

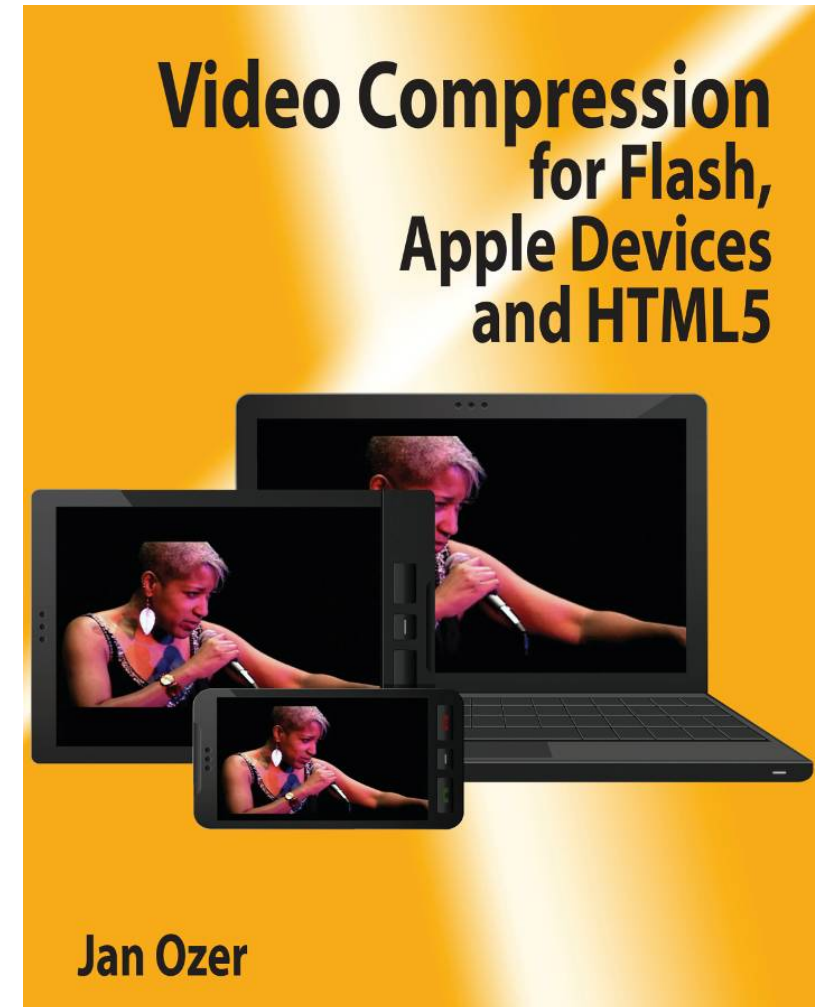


- Creating a multi-file encoding workflow in Telestream Episode

Questions?



- Chapter 2-4 – H.264 encoding
- Chapter 5 – rez/data rate for live and on demand streaming
- Chapter 6 – iOS, Android, Blackberry Windows 7
- Chapter 7 – Adaptive streaming
- Chapter 8 – Choosing an H.264 encoder
- Chapter 10 – Distributing your video (UGC and OVP sites)
- Chapter 11 - Choosing LSSP, encoding hardware and software
- Chapter 12 – Accelerating encoding on multiple-core workstations



<http://amzn.to/ideviceencode>