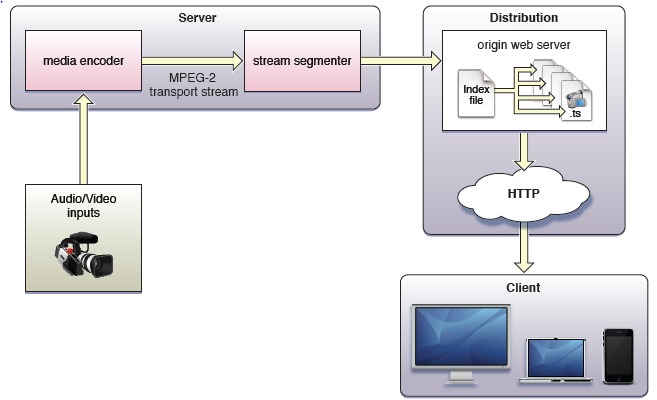
**User Guide for Http live streaming**

**一、Introduction**

**1.** HTTP Live Streaming lets you send audio and video over HTTP from an ordinary web server for playback on iOS-based devices—including iPhone, iPad, iPod touch, and Apple TV—and on desktop computers (Mac OS X). HTTP Live Streaming supports both live broadcasts and prerecorded content (video on demand). HTTP Live Streaming supports multiple alternate streams at different bit rates, and the client software can switch streams intelligently as network bandwidth changes. HTTP Live Streaming also provides for media encryption and user authentication over HTTPS, allowing publishers to protect their work.

**2.**  HTTP Live Streaming sends audio and video as a series of small files, typically of about 10 seconds duration, called media segment files. An index file, or playlist, gives the clients the URLs of the media segment files. The playlist can be periodically refreshed to accomodate live broadcasts, where media segment files are constantly being produced. You can embed a link to the playlist in a webpage or send it to an app that you’ve written.

二. **HTTP Streaming Architecture**



**三、Create http live streaming server**

1、Download the http live streaming tool to windows OS:

Get the package from: <http://172.16.123.146/httplive/HLS_tool.zip>

2、Uncompress HLS\_tool.zip. There are four tool, such as ffmpeg.exe、neroAacEnc.exe、segmenter.exe、x264.exe。

3. **Here is how I have created my streams in a Windows environment:**

（1）Extract WAV file from original video:

ffmpeg.exe -i INPUT.MP4 -vn -acodec pcm\_s16le AUDIO.WAV

（2）Produce an AAC file:

neroAacEnc.exe -cbr 64000 -he -if AUDIO.WAV -of AUDIO.AAC

（3）Produce the required audio-only MPEG2-Transport Stream:

ffmpeg.exe -i AUDIO.AAC -vn -f mpegts -acodec copy AUDIO.TS

(4) Go ahead and segment the audio-only MPEG2-Transport Stream (this can be done later):

segmenter.exe AUDIO.TS 10 Kauai\_HLS/AUDIO Kauai\_HLS/AUDIO.m3u8 <http://172.16.123.146/http/clips/>

Note that this will dump AUDIO-\*.ts and AUDIO.m3u8 files into the .\Kauai\_HLS directory which must exist prior to running.

(5) Choose a bitrate, and encode the video for one of those bitrates. Apple recommends 96 kb/s, 256 kb/s, and 800 kb/s for video. This example will use 96 kb/s (“low” bitrate):

x264.exe --level 30 --profile baseline --bitrate 96 --keyint 30 -o LOW.MP4 INPUT.MP4

(6) Mux the audio and video together to produce an MPEG2 Transport Stream:

ffmpeg.exe -i LOW.MP4 -i AUDIO.AAC -f mpegts -vcodec copy -acodec copy -vbsf h264\_mp4toannexb VIDEO-96KB.TS

(7) Segment the stream:

segmenter.exe VIDEO-96KB.TS 10 Kauai\_HLS/LOW Kauai\_HLS/LOW-96KB.m3u8 <http://172.16.123.146/http/clips/>

(8) Repeat steps 5-7 for Medium (256 kb/s) and High (800 kb/s) bitrates.

(9) upload Kauai\_HLS folder into <http://172.16.123.146/http/clips/>

### 4. Create a variable rate HTTP stream

Once you have creating a single stream down you need to try out creating a variable bitrate stream. There isn't much to it, just create different bitrate encoded streams and link to their stream definition files in a separate stream definition file. Here is an example:

[root@video-machine kauai\_HLS]# cat kauai\_video\_play.m3u8

#EXTM3U

#EXT-X-STREAM-INF:PROGRAM-ID=1, BANDWIDTH=256000

http://172.16.123.146/http/clips/kauai\_HLS/Medium\_256KB.m3u8

#EXT-X-STREAM-INF:PROGRAM-ID=1, BANDWIDTH=800000

http://172.16.123.146/http/clips/kauai\_HLS/High\_800KB.m3u8

#EXT-X-STREAM-INF:PROGRAM-ID=1, BANDWIDTH=96000

http://172.16.123.146/http/clips/kauai\_HLS/LOW\_96KB.m3u8

#EXT-X-ENDLIST

**5、Create html to playback m3u8**

Here is an example:

[root@video-machine kauai\_HLS]# cat videoplay.html

<video src="http://172.16.123.146/http/clips/kauai\_HLS/kauai\_video\_play.m3u8" controls autoplay ></video>

### 6、Prepare the HTTP server

Modify linux Apache server httpd.conf file:

Add below sentence into httpd.conf

AddType application/x-mpegURL .m3u8

AddType video/MP2T .ts

### 7、Test the stream

(1)、connet wifi 9L02 with PR3.The build of PR3 must support http live streaming.

（2）、open browser and input "http://172.16.123.146/http/clips/kauai\_HLS/videoplay.html"

（3）、go

**四、http live streaming encrytion stream**

**1、Encrypt ts streaming via openssl**

Script:

#!/bin/bash

openssl rand 16 > static.key *#generate key file(a packed array of these 16 octets in binary format)*

key\_as\_hex=$(cat static.key | hexdump -e '16/1 "%02x"') *#read pass phrase from key file*

for i in {0..1}; do

init\_vector=`printf '%032x' $i`

openssl aes-128-cbc -e -in kauai\_High\_800KB-$(($i+1)).ts -out kauai\_High\_800KB\_enc-$(($i+1)).ts -p -nosalt -iv $init\_vector -K $key\_as\_hex

*# encrypt ts files*

done

**2、Edit index file**

e.g.

[root@video-machine kauai\_HLS]# cat kauai\_High\_enc.m3u8

#EXTM3U

#EXT-X-TARGETDURATION:10

#EXT-X-KEY:METHOD=AES-128,URI="static.key"

#EXTINF:10,

http://172.16.123.146/http/clips/kauai\_HLS/kauai\_High\_800KB\_enc-1.ts

#EXTINF:10,

http://172.16.123.146/http/clips/kauai\_HLS/kauai\_High\_800KB\_enc-2.ts

.

.

.

.

http://172.16.123.146/http/clips/kauai\_HLS/kauai\_High\_800KB\_enc-18.ts

#EXTINF:10,

http://172.16.123.146/http/clips/kauai\_HLS/kauai\_High\_800KB\_enc-19.ts

#EXT-X-SENDLIT

**3、Create html to playback m3u8**

e.g.

[root@video-machine kauai\_HLS]# cat test\_enc.html

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8">

</head>

<body>

<div style=" text-align:center">

<video src="kauai\_High\_enc.m3u8" controls autoplay ></video>

</div>

</body>

</html>

### 4、Test the encrytion stream

(1)、connet wifi 9L02 with PR3.The build of PR3 must support http live streaming.

（2）、open browser and input "http://172.16.123.146/http/clips/kauai\_HLS/test\_enc.html"

（3）、go