

Techniques of Streaming Stored Multimedia

By

Shreshtha Shyamsundar, Kamalkumar Rathinasamy

Infosys Limited, Mysore, India - 570027

Infosys Limited, Mysore, India - 570027

Shreshtha_shyamsundar@infosys.com, Kamalkumar_r@infosys.com

ABSTRACT

Inline to the growing demand for information on the web, streaming of data has gained much importance across tracks; however at the same time so has evolved the methods towards plagiarizing the content sent across using complicated techniques. This article talks about the ideas towards how one could protect the streamed across stored media from being stolen yet leverage the potential of entertainment that it could generate to the best extent possible.

General Terms

Multimedia streaming, Stored audio streaming.

Keywords

Streaming, Audio content streaming, Multimedia streaming.

1. INTRODUCTION

Latest advances in computing technologies, storage devices and high-speed networks have made it viable to stream content effectively for a fraction of the cost compared to those from yesteryears. "Stored media" refers to content that has been previously recorded and encoded.^[1] Apple's iTunes Music Store serves as an example of a centralized stored media system^[1]. Scalable on-demand streaming of stored media can be achieved using scalable server protocols such as patching^[2] and Hierarchical Stream Merging^[3], server replication as with CDNs, and peer-to-peer techniques. Needless to say, the above strategies work best when the connections are reasonably stable. This article primarily focusses on enabling streaming stored audio content and how can the content be best protected from the plagiarists.

2. ORCHESTRATION – AUDIO STREAMING

The below architecture diagram principally illustrates the different points involved in the stored audio being streamed across from the server to the client. Broadly the scheme is divided into areas viz.

- Stored Audio Collection
- Preparation of Media Content
- Streaming (Client & Server)

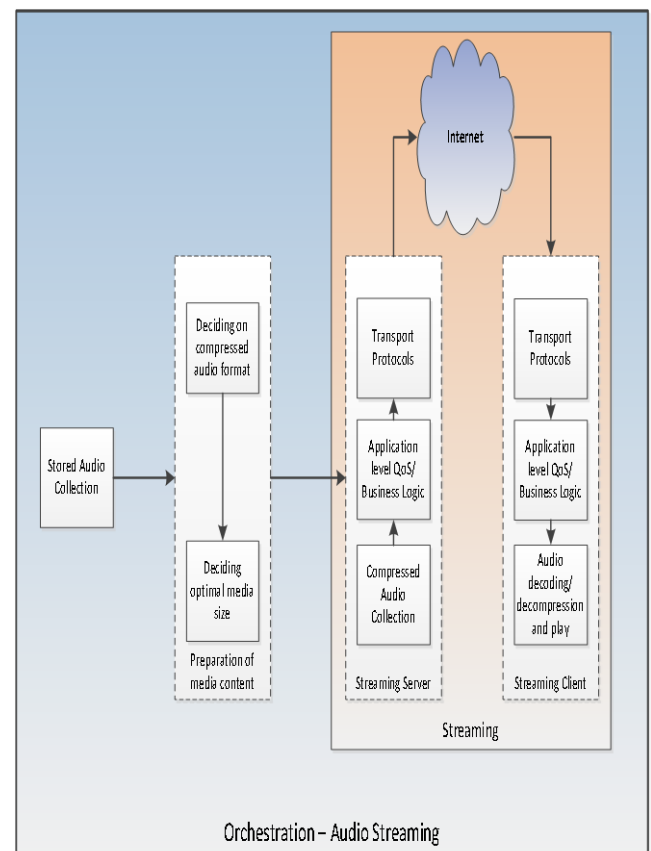


Figure1: Architectural flow – Media Content Streaming

2.1 Preparing media content

Although music composers and aficionados are not at peace with the proliferating acceptance of the compressed audio files particularly given the effort put in by them towards striving for sonic excellence, it is now a globally acknowledged fact that compressed audio is here to stay. Owing to this, personal audio players such as the iPod from Apple, Zune from Microsoft, and Walkman from Sony have come up with portal audio players that can hold up to thirty-five thousand four minute audio files in comfort. The process of compression employs a combination of digital technology and the science of aural perception (psychoacoustics) to remove data bits from the original digital file that are considered to be inaudible. These bits can include frequencies

beyond the normal threshold of human hearing, sounds that are masked by other sounds, and various other “redundant” sonic information.^[4]

The point of contention with this whole concept is just how much of that data is truly inaudible. While some bits can be removed with little consequence, much of what gets stripped away can subtly affect our perception of how things sound. While moderately compressed files can deliver near-CD quality sound, too much compression can remove elusive qualities that can make a difference to how we perceive music on a subconscious level.^[4]

2.1.1 Compressed audio formats

While MP3 is a widely accepted standard in terms of the compressed audio format; there are other formats that work on discarding the unnoticeable data portions of the file to keep the quality intact. Some of these formats are discussed below:

WMA (Windows Media Audio) was created by Microsoft, and is often offered as an alternative to MP3 on music and video download sites. It has also common on sites that offer streaming audio and video files compatible with Windows Media Player. While many people feel the sonic quality is superior to MP3, WMA files tend to sound overly bright and brittle, with less than optimal stereo imaging.^[4]

AAC (Advanced Audio Coding) is the default standard for iTunes, the iPod, the iPhone, as well as PlayStation and Nintendo DS. It's also often used as the audio component for Apple's QuickTime and MP4 video formats. Generally speaking, if you are going to offer a second encoding format online, AAC is the one to consider.^[4]

AC3 is a format developed by Dolby and is often used for video soundtracks due to its superior stereo imaging and ability to handle multitrack formats like 5.1 surround. Because of this, many consumer-grade DVD players support AC3 format.^[4]

RA (Real Audio) is a fairly good-sounding coder-decoder, but is on the decline because the files only play on Real Audio's proprietary player. Many people refuse to install the Real Audio player due to its excessive advertising and high-CPU demands.^[4]

Ogg Vorbis is an open standard audio format that delivers a very high-quality sound. Unfortunately, like many open standard projects, it has had a hard time catching on among users.^[4]

FLAC (Free Lossless Audio Codec) is one of the few audio formats that deliver truly lossless compression. FLAC is similar to a ZIP file, but is designed specifically for audio. FLAC is also open-source, and FLAC files can be played back on most MP3-compatible players. (There are several other lossless formats, including Apple Lossless, WMA 9 Lossless, Monkey's Audio and MPEG-4ALS, but none offers the open compatibility of FLAC.)^[4]

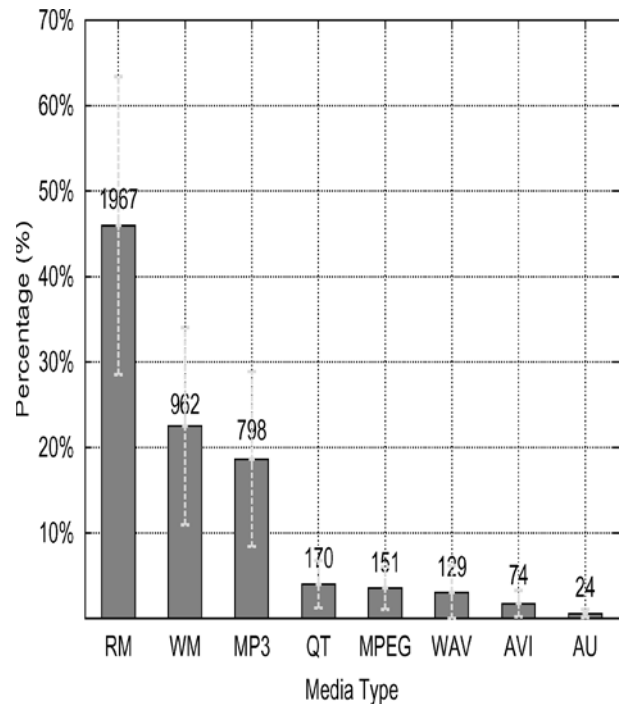


Figure 2: Popularity in terms of acceptance rate of different audio formats^[5]

2.1.2 Deciding Media Size

The original media file size has to be reduced to the best extent possible with no noticeable compromise on its quality. A media file's quality is determined by an attribute entitled, bitrate, which also determines its size. Reducing bitrate in turn reduces the file size. Although there is no standard optimal bitrate for streaming media files; the optimal bitrate for each media file is usually arrived at after trials. The bitrate required for a non-music audio file, i.e., audio books, taped speeches, and would be much lesser than the bitrate required for music audio file.

Should the need arise to support both streaming and downloading options, then usually two copies of the media files are stored at the server, one with a reduced bitrate (streaming) and the other at the original bitrate (downloading). The web request for streaming can be served with the reduced bitrate media file, and the web request for downloading can be served with the original bitrate media file. There are websites where multiple options are provided to the users to choose between downloading media files with varied bitrates. Any user who is prepared to wait a little longer to download but can enjoy high quality media file offline can opt the high/original quality bitrate.

Several bit rates are specified in the MPEG-1 Audio Layer III standard: 32, 40, 48, 56, 64, 80, 96, 112, 128, 160, 192, 224, 256 and 320 kilobits/s, with available sampling frequencies of 32, 44.1 and 48 kHz.

Uncompressed audio as stored on an audio-CD has a bit rate of 1,411.2 kilobits/s, so the bitrates 128, 160 and 192 kilobits/s represent compression ratios of approximately 11:1, 9:1 and 7:1 respectively.^[6]

2.2 Streaming Stored Audio

Music composers invest billions of dollars in creating world class audio content therefore it becomes crucial for the industry to take steps towards stopping the distribution and creation of the plagiarized and pirated content thereby protecting the intellectual property.

For instance, the story of Napster where the website helped facilitate the transfer of copyrighted material that was later deemed unlawful in 2001 using Peer-to-Peer (P2P) sharing helps underscore this fact.

Below is an illustration of the generic process of working of how stored audio is streamed across from the server to the client with the assistance of a media server

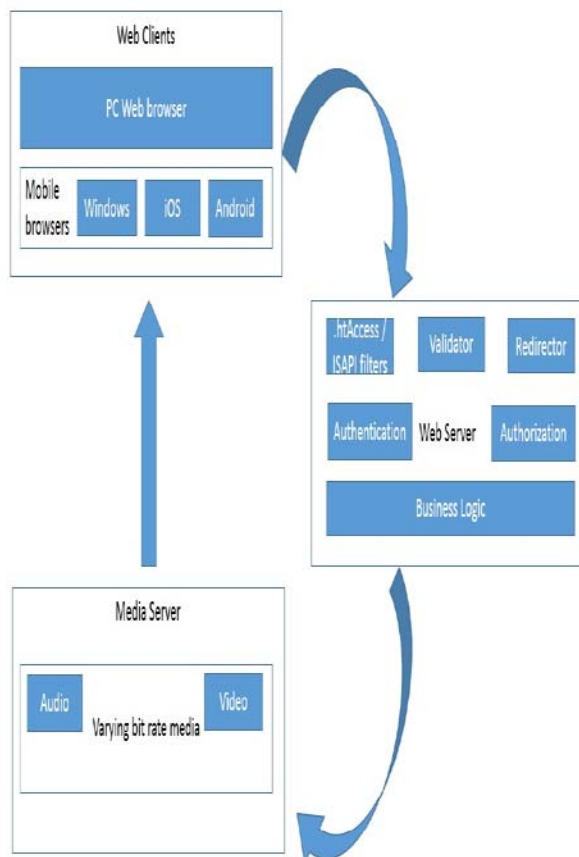


Figure 3: Streamed Audio – Technology Simplified

2.2.1 Web request for content

When a client requests for content either via streaming or download of media i.e. images/audio/video; the request is directed to a particular folder on the web/application server, say *ExposedFolder*.

Fundamentally, the web call needs to contain both the media filename apart from the session identifier to identify the transaction uniquely within the session for honoring the request.

2.2.2 Validation of the Incoming request

As discussed in the last step, apart from uniquely identifying the transaction, the flow has to validate the request towards checking for authorization and authenticity.

Perform the following validation on all requests to contents in *ExposedFolder*.

- Check if the user has logged in by validating current session key (passed as request parameter).
- Check if the user has permissions to access the media content.

Next step is to consider ways to inject the validation code before the media files are sent to web client.

2.2.2.1 Request to validator module

One option is to develop the web client such that the request is sent not for the media file directly, but for a server program module (along with arguments containing the media file information), which validates and sends the required content back as a response. Most of the standard native (javascript) web clients / web media players expect the server request URL to end with the standard media file extensions. If the server request URL is not for a media file (if it does not end with media file extension), most of the players do not treat the response as media content regardless of the response MIME type.

2.2.2.2 Request to proxy media location

An alternate way to inject the validation code is by either filtering the call or by applying per-directory access control. These strategies are web server dependent.

2.2.2.2.1 Hypertext access (htaccess)

Web servers like Apache Web Server, Oracle iPlanet Web Server and Zeus Web Server support a directory-level configuration file called *.htaccess* (hypertext access).

The hypertext access file is about allowing access control at the directory level. It can be configured such that all requests to the particular directory of media files gets redirected to the validator server module, where the request is verified based on the configured rules and responded back with the required media content.

There is a couple of ways how the validator can respond back with the media file content.

2.2.2.2.1.1 File read, store and send response

The default option is to read the media file content and pass it as a response by mentioning the appropriate 'Content-Type' in the response header. The limitation with this approach is that web servers usually will have some limitation on the file size that can be read and stored in its memory and hence this approach might suite only for requests for smaller sized media files.

2.2.2.2.1.2 Redirection

An alternative approach for the validator to respond back with the media file content is to redirect the request to another directory location containing the actual media files, say *ActualFolder*. Since the operation is a server redirect, there are no limitations on the media file size (the memory module does not contain this information). The *ActualFolder* will not have any validations as part of its hypertext access file. Any user who browses through the web client HTML code or who monitors the web request URL would not know the actual file location, and so there is no way the user could directly access the media file without an approval from validator server

module. However, any user who monitors the response file would see that the response URL is different from the request URL. The response URL will be pointing to the *ActualFolder* location and not the proxy location, *Exposed Folder*, specified in the call. Therefore, if the user tries to access this response location directly from web browsers, this call would fetch the respective media file directly bypassing the validator module that would pose a significant security risk.

However, this security vulnerability can be addressed by tweaking the response headers. Tweaking response headers is a web server specific solution. Though all modern browsers honor the response headers to an extent, it is not guaranteed that they would honor all the response headers.

2.2.2.2 ISAPI filters

IIS Web Server Web servers like IIS can make use of ISAPI filters to inject the validation code.

Similar to having the hypertext access files on the apache servers, Microsoft's web server; the IIS (Internet Information Services) has a component entitled ISAPI – Internet Server Application Programming Interfaces which can be used to be able to modify incoming or outgoing data calls with ease.

The operation of the ISAPI is based on having these filters registered with the IIS that helps redirect the calls coming into IIS onto a script which can be used for any particular processes validation, authentication and authorization. Post the pre-processing, the request is serviced failing which errors can be aptly handled.

3. SECURITY ASPECTS – CACHING AND STORING

Any streaming media that is delivered to the client is stored at a browser specific location in the client computer. User can access that location and access the media file. One must take proper care towards ensuring that the data does not get stored at the temporary locations on the browser particularly while streaming media content.

One way to not to save the entire file at the client location is to stream only the required chunks of the media file and flush them out once the new set of chunks reach. Consequentially at any point of time, only the minimal set of chunks would reside in the browser cache. It is recommended to set correct caching options in the response header to prevent the web pages from storing old chunks of media content on the client computer. The headers are however browser dependent. In general, the `no-store` header would work on the majority of the browsers.

Even if the media file has to be streamed in chunks, the actual source media file can reside in the server as a single file. There are methods available to generate chunks on the fly and stream it to the web client thereby eliminating the need to maintain more content at the server.

4. ALTERNATIVE SOLUTIONS TO STREAM CONTENT

The idea behind highlighting other ways to enable the audio streaming was to contrast other solutions equally preferred in order to guide the developer in terms of the best fit solution.

4.1 Dynamic URL

One of the reasons why streaming media is vulnerable is that the user can retrieve the source URL from the web page source code and place subsequent requests for the same media file. One solution is to build dynamic URLs (like `media.avi?id=1asd2sd5sdlj4`) where this id would be valid only in this session. Hence, storing this URL does not yield any benefits to the end user.

4.2 Packaging & Licensing

No matter how the media files are streamed to the client, there are many tools available that can save the media files to the user's local machine. Though it might be impossible to prevent a user from saving the media file, one way out would be to package them and send the license file for playing the packaged media file. Packaging recommends encrypting all media files and modifying the content header to contain a redirect URL. If user downloads the packaged file and tries to play the same and if the license is not found, the client media player will display the redirect URL requesting the user to obtain a valid license. This way, even if the media file is shared with others, they cannot be played due to the license file dependency.

4.3 Custom Player

The point this article means to address is about creation and use of a custom format and custom player to play it; the chances for other off the shelf players leveraging content of this form would be remote.

a) Custom Format

Creating custom media format and developing custom web player to play that custom format is one way to protect the media files. In the worst case, if it is intercepted, it cannot be comprehended by any other media player. Though it is technically possible to develop custom format and player, it is complex, as it involves coming up with a proprietary media encoding and compression format.

b) Encryption

An alternate to creating a custom format is to encrypt the media file with any appropriate key based encryption algorithm. Custom media web players can be developed to decrypt the media file. In this case, the media file cannot be read as it is, by any other media player.

4.4 DRM

DRM i.e. digital rights management follows a pattern that is usually used by publisher and copyright holders to be able to protect their investment in terms of intellectual properties from being misused or abused against the purpose intended. This pattern is one of the most restrictive in nature by virtue of allowing control on pretty much any/all the operations that a user can do with the content/document protected by DRM.

Digital Rights Management Techniques include:
(1) *Restrictive Licensing Agreements: The access to digital materials, copyright and public domain is controlled. Some restrictive licenses are imposed on consumers as a condition*

of entering a website or when downloading software.^[7]
 (2) Encryption, Scrambling of expressive material, and embedding of a tag: This technology is designed to control access and reproduction of information that includes backup copies for personal use.^[7]

5. CONCLUSION - FUTURE OF MANAGING STREAMING MEDIA

The model that existed before seemed very modest for e.g. the theatres would make money through selling tickets, sale of wares happened from across the counter and people would walk to shops to make a purchase. The rationale was always to get the customers to visit the stores.

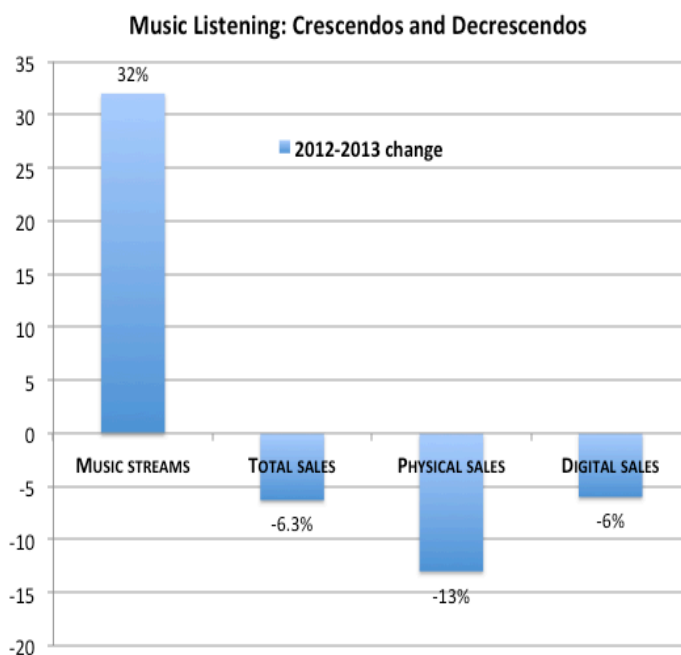


Figure 4: Ascent of Streaming content versus alternative mechanisms^[8]

Streaming is the future, but it does not yet dominate the present. Online music is a six billion dollar business. Subscription streams, which have tripled in the last three years, only account for \$1.1 billion of it.^[8]

Essentially the Online business in Netflix, Lovefilm® is in a model that's here to stay. Improvements in the technology space have substantially aided the cause.

a) Bridge the gap in quality – Earlier it was just the quality of the broadcast media which used to be of HD quality in terms

of surround sound and visuals – however the streaming audio quality with the tech improvements have improved leaps and bounds in terms of quality.

b) Enhanced focus on mobile as a media to stream content – By virtue of allowing content on the handheld media, the need of being available to listen to a song or watch a show at a particular hour is eliminated thereby increasing viewership

With the proliferation of the online streaming of stored audio, there is a definite threat of piracy in terms of being unable to secure and protect the content completely from everyone however the effort remains to keep it tight and locked away from the vast majority of folks that access it.

This paper attempts to address the portion of securing audio content with the best possible means currently available and implemented by the authors.

6. REFERENCES

- [1] Kyung Wook Hwang, Vishal Misra and Dan Rubenstein. (N.d.) Stored Media Streaming in BitTorrent-like P2P Networks
- [2] Eager, D. L., Vernon, M. K., Zahorjan, J.: Optimal and Efficient Merging Schedules for Video-on-Demand Servers. In: Proc. ACM MULTIMEDIA '99, Orlando, FL, Nov. 1999, pp. 199--202.
- [3] Eager, D. L., Vernon, M. K., Zahorjan, J.: Optimal and Efficient Merging Schedules for Video-on-Demand Servers. In: Proc. ACM MULTIMEDIA '99, Orlando, FL, Nov. 1999, pp. 199--202
- [4] Understanding Audio Data Compression: MP3, AAC and more (25-04-2011). Retrieved on May 16, 2014. Available at : <http://www.uaudio.com/blog/understanding-audio-data-compression/>
- [5] MINGZHE LI, MARK CLAYPOOL, ROBERT KINICKI, and JAMES NICHOLS (Jul 2004) - Characteristics of Streaming Media Stored on the Web
- [6] BitRate defined at Wikipedia (n.d.) Retrieved on May 16, 2014. Available at <http://en.wikipedia.org/wiki/MP3>
- [7] Wikipedia – “Digital Rights Management” (n.d.) Retrieved on 16 May 2014. Available at http://en.wikipedia.org/wiki/Digital_rights_management
- [8] The Future of streaming media (April 10 2014) – Derek Thompson. Retrieved on May 16 2014. Available at <http://www.theatlantic.com/business/archive/2014/04/the-future-of-media-will-be-streamed/360473/>