

# MULTI QUALITY VIDEO-ON-DEMAND

Riccardo Petrocco

st. #1339702

## Introduction

In the current existing peer to peer systems, a Video on demand functionality is actually impossible to handle correctly for the majority of the users. This is caused by the fact that, to avoid free-riders, every peer could download only as much as he's uploading, especially for ADSL users where the available bandwidth will drop down to the lower level given by the upload capacity, normally around 1/5 of the original capacity. This explains why with the current methods watching a movie without downloading it all is trivial task. It is also not a good idea to create a video with a lower quality because in fact we cannot know the available bandwidth of a user, and even if we could know the type of connection of a peer, the available bandwidth will change all the time, depending on different factors like the number of available seeds, the amount of bandwidth used by other programs or the speed increasing of active downloads that are not determinable.

The multi bit rate VoD technique should solve this problem by changing the quality of the video while downloading it depending on the available bandwidth.

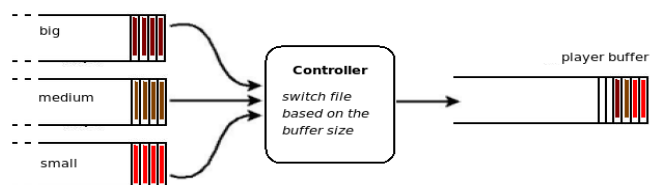
## Creating the video

The idea is that from a video file (could be of any kind), through an encoder algorithm three or more videos are created, with different qualities and of course different sizes, and from those a new torrent is created containing actually 3 different versions of the same video. This torrent will be then available for normal bittorrent clients, in our case the "tribler" program.

## Downloading the torrent

The client will then download the torrent file filling different buffers depending on the quality, in my case three buffers, (or just storing the location of the downloaded pieces). A controller will then be responsible of handling the buffers sending the right pieces of video to the player.

Based on the current buffer size the controller will try to switch to a higher video quality if there is enough bandwidth available and if the higher quality buffer is already filled up with the correct pieces of video.



The controller will tell the bittorrent client to download only the pieces needed for the current playback. Only the lower quality video will be always downloaded to assure a continuous playback even if connection drops down or if some unexpected situation occurs, like having seeds disconnecting from the P2P network.

## Reproducing the video

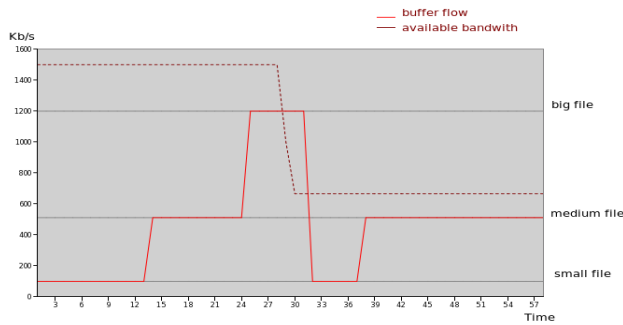
The problem of reproducing the video smoothly is that there is the need of some kind of synchronization between the different quality levels. It should be possible to switch the video quality having a continuous playback and more important when switching to another file we should start with a new Group of Pictures<sup>1</sup> (GoP) having at the beginning

---

<sup>1</sup> GOP is a group of successive pictures within an MPEG-coded video stream

a clear image (an I-frame<sup>2</sup> considering normal encoder algorithms). Depending on the conventions used for encoding, the I-frames will be located with a different frequency in the video depending on the quality, switching at a particular time to another video without losing frames or without a synchronization technique is almost impossible. This is given by the fact that every encoded video has a certain amount of clean pictures (frames) that are used as reference for the motion frames, more the video is compressed, lower the quality is and less information will be stored in the motion frames.

To correctly handle the synchronization between the files I decided to split every video file in the same amount of



independent pieces of fixed time length encoded separately. In this way every encoded piece will start with a new clear frame, and a good synchronization technique, based on time and not on Group of Pictures, is offered.

After some investigation I have noticed that it is not possible to avoid having the player read codecs information when switching to a different quality without having big limitations on the kind or codec to be used. What is possible to do is to decrease the time needed for switching to almost imperceptible level for

humans, normally for the audio track is something like less than 6 ms, when changing quality.

Currently I am working on trying to merge video pieces of the same quality level, encoded in the same way, in a transparent way for the player. In this scenario we will have a flicker free playback for the same video quality file and a small hanging when changing from one quality to another. This is acceptable considering that the video quality will change only during the first 5-20 seconds, until the higher possible quality for the specific bandwidth is reached, or in case of a connection problem that can be caused by different factors.

---

2 Each GOP begins with this type of picture. It's the reference picture independent from any other motion pictures