# ABR (Adaptive Bit-Rate) in the Cloud

## A new approach to the Adaptation Algorithm

**Bharath Madhavan[1] and Murugesan Nallathambi[1]**

**[1] Technical Leader,
SPVTG (Service Provider Video Technology Group)
CISCO Inc., Chennai, Tamil Nadu, India,**

### Abstract

As of most well-known ABR (Adaptive Bit-Rate) technologies, the complete algorithm that handles the adaptive bit-rate algorithm resides entirely in the individual clients and the clients “Adapt” based on the network congestion that each one of them observe. But there is no “overall” picture that a client can have with respect to the network behavior. The proposal here is to move some of the adaptation logic to the “cloud” where it is possible to have a global picture of all the clients that are currently streaming content within the service provider network and so it can turn out to be a better overall solution and a better streaming experience for all the clients within the network.

***Keywords:*** *Cloud computing, streaming, ABR, HLS, DASH, DRM, Security, Web services*

## 1. Introduction

This paper looks into the possibility of a new approach to the current Adaptive Bit-rate technologies with a focus on moving the adaptation logic to the cloud instead of keeping it within the clients as it currently exists. The paper explains the current ABR model and tries to identify some of its obvious limitations and see how these can be resolved as a new proposal.

## 2. Current model and its limitations

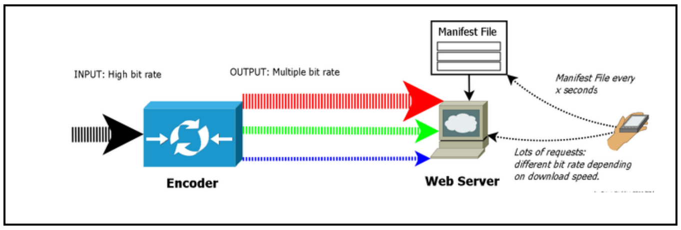
The first thing to do is to study the existing ABR technology and how the current Adaptation logic works.

### 2.1 What is ABR

ABR stands for Adaptive bitrate streaming technology and this is a method of video streaming over HTTP where the bitrate and resolution of the video content can be increased or decreased in real time based on the user’s network bandwidth and his device’s CPU, ensuring him the best possible video quality. The player handles the complexity of changing bitrates so there is no user intervention. Let us take a look at ABR technology here, just to observe some of the key factors of ABR.

### 2.2 Why ABR

- End users always get highest quality content based on their network bandwidth.
- They will not see video buffering delays as long as they have minimum bandwidth requirements of the video content.
- No user intervention needed to switch from one bitrate to other.
- As the streaming is HTTP based, no special streaming servers are required for the video publisher. A simple HTTP server would suffice.
- User can seamlessly switch between Wi-Fi and 3G without any visible jitters.
- Intelligence is moved from network to client.


Manifest File
INPUT: High bit rate
OUTPUT: Multiple bit rate
Manifest File every x seconds
Lots of requests: different bit rate depending on download speed.
Encoder
Web Server

### 2.3 How ABR Works

Fig. 1 ABR data flow model

- The AV source is encoded into multiple bitrate files.
- Each bitrate file is split into multiple short segment files of 2 to 4 seconds long approximately. To enable

this splitting, the encoding is performed with specific guidelines like closed, fixed length GOP, etc.

- Splitting along GOP ensures frame alignment across various bitrates. Each segment can be decoded without any dependency on other segments.
- Special post-processing tools will be used for splitting the source file into many small "chunk" files.
- A manifest file is constructed which contains information like available bitrates, URIs of the chunk files, Encryption information etc.
- The ABR client residing in the user's device (could be a set-top box, tablets or smartphones) will access this manifest file and start downloading the lowest bitrate segment.
- If the client finds that the download speed is greater than the bitrate of the segment, it will request for the next higher bitrate for the next segment. Continuing this process, the client will lock to the best bitrate for the device. It can also factor in the device's CPU capabilities for finding the best bitrate.
- Client handles all the logic which is transparent to the end user who will see continuous video except for some change in quality.

With this brief introduction to ABR technology, it is clear that all the logic to adapt to changing network bandwidth is within the client and it is only the client who decides the bitrate of the video segments to be downloaded. As it was observed, it is actually one of the advantages of moving the bitrate selection intelligence from the network to the clients in a sense because it is easy for the client to adjust itself and adapt based on the congestion it sees in the network.

However, there are some pitfalls to this approach of having the adaptation logic within the clients, some of which are mentioned here below.

## 2.4 Greedy Clients

Every client has one big disadvantage which is that it does not know about the existence of other clients in the network. Given this limitation, the client model can be implemented in only one way, which is to be done in such a way that the client gets the best bit-rate as is available for itself.

However, when you consider a service provider scenario offering IP video services for instance, there will be several clients / subscribers all using these TV services and thus all of them utilizing the bandwidth of the same network on which they are all streaming the video content.

Consider a very simple example of say 5 clients streaming ABR content to their respective homes from the network and say they are steadily streaming content say for 480P resolution bit-rate. Now, take the case where due to some relief in some network congestion, some bandwidth has been freed up in the network so that the download of these ABR chunks happens faster. This immediately triggers the ABR logic running in the client(s) and all these 5 clients running this logic calculate the available bandwidth and feel that they can now scale over to the next higher bitrate to support say 720 resolution.

Now, what happens is that all these clients are going to go ahead and try to grab this extra bandwidth that has become available on the network. That is, the clients are greedy that whenever they see any extra availability in network bandwidth, they make the assumption that this bandwidth has to be utilized by them and they can go ahead and download the higher bitrate file segments.

Even though it shows as if the clients are greedy, the actual underlying cause for this problem is that the clients are not greedy but they are ignorant of a global overall view of the network but they can only think about themselves individually. So, they try to grab all that they can from the available bandwidth.

## 2.5 Uneven Distribution of Video Quality

Another problem of this ABR model is that it can result in degradation in performance on some clients versus the other clients and it may result in an uneven distribution of quality video across all clients.

Going back to that example mentioned above, just as soon as some bandwidth frees up in the network, the clients are going to recognize this and recalculate their new bit-rate. But, it is not always the case that all the clients are going to run the adaptation logic at the same time. Say, client 1 and client 2 did the calculation first and since they determined that they can increase their bitrate, they move to 720P resolution. But, by the time the other clients come around to doing their adaptive calculation, it is already too late as the first two clients have already started downloading content at a higher bit-rate and so the originally available extra network bandwidth is already consumed and it is no longer available anymore for the other clients. At this stage, it is also possible that one of the other clients will have to downgrade to a lower bitrate because the network bandwidth has been taken up more by the first 2 clients.

This is displayed in the figure below. What started out as all the 5 clients streaming at 480P, an intermediate increase in network bandwidth has resulted in an uneven distribution of 2 clients at 720P, 2 clients at 480P and one client downgraded to 320P. This is an inherent problem of the existing ABR adaptation model which is shown below.

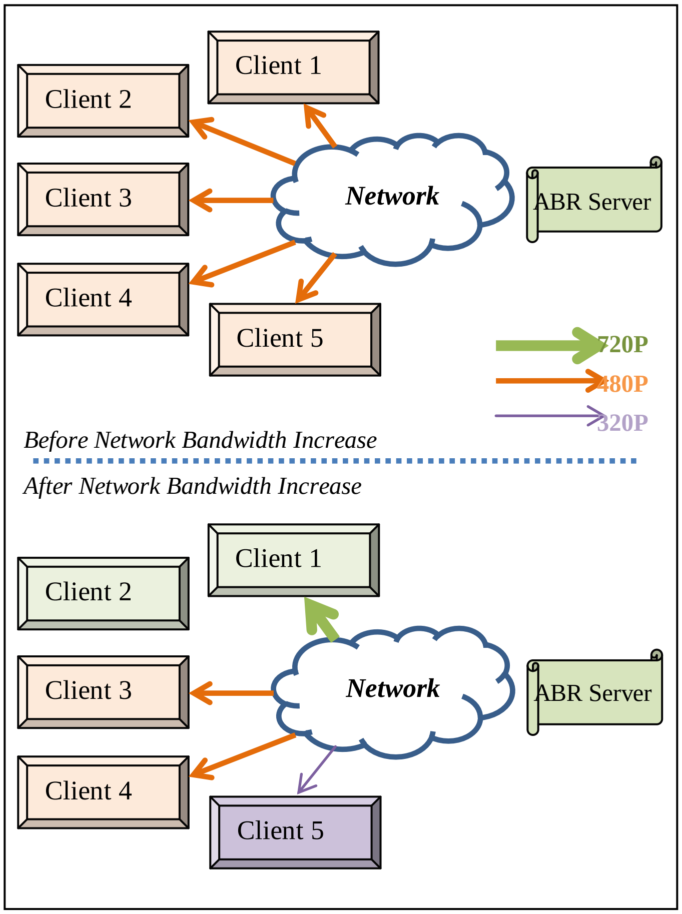


Fig. 2 ABR Quality Impact due to Network Bandwidth Change

## 2.6 Sine Curve

Another problem with this traditional ABR client logic model is the regular expansion and contraction of network bandwidth consumption by the clients. This again is a direct effect from the concept that the clients do not know each other’s bandwidth consumption details and their algorithm is solely based on their own bandwidth consumption only.

Consider this following scenario, very similar to the previous example discussed. When a little bit of network bandwidth got freed up for use by the ABR clients, all the clients are going to upscale their bitrate calculations and start downloaded higher bitrate content files. This will obviously result in the over-utilization of the network bandwidth and so, the clients will now have to “adapt” to this loss in network bandwidth. Not knowing that they themselves (in a collective sense as a group of clients) are the cause for this network congestion, all the clients will now try to reduce their bitrate usage. As this happens, again, network bandwidth will get freed up. Thus, this sine curve of network bandwidth utilization happens.

Now, this cycle of under-utilization and over-utilization of network in a cyclic sine curve is possible that all the clients are caught up in for quite some time. However, it is also possible a steady-state may be reached after some time and these oscillations may stop. Again, a different network utilization condition may again trigger these sine curve oscillations, and again it may take some time to stabilize and this can go on forever on the network. A perfect steady state is impossible as the dynamics of the network are so intense and the clients are going to continuously keep these waves of ups and downs in the network bandwidth utilization.

The only way this problem can be resolved, or atleast brought to a decent control is when there is an overall view of the network characteristics that can run as a “Master ABR Algorithm” in a central location which can monitor this overall behavior and apply adaptive logic to all clients simultaneously so as to keep these fluctuations under control. It may seem better to maybe under-utilize the network bandwidth a little rather than be on the verge of full usage which results in these fluctuations in bandwidth usage.

### 2.7 Unmanaged ABR Algorithms

When one hands over complete control of the entire ABR algorithms to clients, a lot of discrete, unmanaged, and rogue ABR algorithms may become possible to run on clients and can cause great impact to the proper functioning of the network. This is a big problem for Service Providers who will not have any control over the functioning of ABR clients within their network

Basically, Service Providers who provide premium content are not able to keep a control of the bit-rate at which the clients download their content. ABR client logic only says, as long as a client has enough bandwidth for himself, he can go ahead and download the highest possible bitrate content for himself, no matter how this affects the network or the other clients in the network. This affects the quality of service offered by the Service Providers to the clients at large and unfortunately, they are not in any position to control the same.

## 3. Benefits of moving to the Cloud

For all the problems mentioned above, there is a need to look for ways to solve them. One possible way that will be addressed in this writeup is to take some of the adaptation logic and move it to the cloud. So, instead of the entire decision making logic being on the individual clients, the proposal here is to break this logic into two parts, Master logic and Slave logic, and have the Master logic in the cloud and keep the Slave logic in the clients as before.

Before looking at how this concept will work, let us look at the advantages of taking this approach. There are several advantages of taking this approach. They are:

- Being in a centralized location, the Master logic module can determine more accurately, the “best” bit-rate utilization for the clients.
- The server can through a different path to the clients, find out the bandwidth availability of the clients ahead of time and “predict” any changes in bit-rate utilization of the clients prior to the bandwidth actually going down or up.
- Uneven distribution of bandwidth across clients can be avoided / corrected as the Master agent can see or predict this unevenness and adjust within its algorithm to fix this problem.
- A more conservative approach can be taken by the Master logic module which gives a buffer in the network bandwidth and not always choke the network.
- The service providers can apply various different “licenses / rules” on different clients based on the subscription policies of the clients and the bandwidth utilization rules can be tweaked accordingly by the Master logic module.
- Rogue clients can be kept under check and even blocked if they try to bypass the Master and go directly to fetch content.
- A uniform policy / algorithm can be in place for the entire consumer base of a service provider.
- The service provider can “change” the Master logic module to use different types of algorithms, without impacting the clients.
- This service itself is a value-add to the customer from the service provider’s perspective and it can be charged as a service to the customer who can choose different levels of ABR service as appropriate.
- Redundancy of ABR logic servers is possible.
- Load sharing of ABR logic servers is also possible.

## 4. How does this work

In this section, an initial attempt is made to explain how this model is intended to work. Without getting into too much detail of any possible implementations, let us take a look at a possible model.

The concept is quite simple. There is an ABR Master Logic module running on one of the servers in the cloud. It can be on the same machine which is the ABR server which is also hosting the content. Then, there are the ABR Slave logic modules which are running on the clients. The Master and the Slave communicate over REST APIs (or some other RPC mechanism).

After some initial handshake between the clients and the server, whenever a client wants to play an URL, the URL is given to the Master Logic module in the server which returns back the “Initial bit-rate” information to the client and the client starts streaming at that rate.

Now, there are two situations where the bit-rate for a particular client can change. One is a synchronous path and the other is an asynchronous path. In the synchronous path, as the client is also monitoring for network congestions or improvements from its side, when any change in its download speed is observed, it asks the

Master agent to check if the bit-rate calculation needs adjustment. Now, the Master agent returns back saying if this is possible or not and based on that, a decision to switch bit-rates is made. The other path is the asynchronous path where the Master agent in the cloud, as it is constantly monitoring the network usage and the trending of the bandwidth usage, can asynchronously request client(s) to change their bit-rates in order to adapt to the changing network conditions.

## 4. Conclusions

This “client – server” cloud model for ABR adaptation logic algorithm will solve all the problems mentioned in the previous sections in the existing ABR solutions. A detailed study in this area is in progress and experiments are being conducted to actually see the various benefits that can be derived from this cloud model.

### Acknowledgments

The authors would like to express thanks to the New Initiative team within Cisco who provided certain pointers towards this technology. The authors would also like to express thanks to the management team of Client Software business unit in Chennai for facilitating this effort in working on this paper.

## References

[1] IEEE Project: “P2302 - Standard for Intercloud Interoperability and Federation (SIIF)”, http://standards.ieee.org/develop/project/2302.html

[2] Ania Monaco, “A view inside the cloud”, IEEE Tech Topic, http://theinstitute.ieee.org/technology-focus/technology-topic/a-view-inside-the-cloud

[3] Alexander Pasik, CIO IEEE, “Cloud Computing: The Inevitable Answer”, IEEE Roundup discussion, http://theinstitute.ieee.org/ieee-roundup/opinions/ieee-roundup/cloud-computing-the-inevitable-answer

[4] Ivan Berger, “For Emerging Markets, a focus on the cloud”, IEEE International Conference on Cloud Computing.

[5] Saamer Akhshabi, Sethumadhavan Narayanaswamya, Ali C. Begen, Constantine Dovrolis “An experimental evaluation of rate-adaptive video players over HTTP”

[6] C. Liu, I. Bouazizi, M. Gabbouj, Rate adaptation for adaptive http streaming, ACM MMSys, 2011.

[7] Cloud Security Alliance, https://cloudsecurityalliance.org/

[8] Adam Thursby, “Leveraging the Cloud for Your DVR”, http://dvr.about.com/od/cabledvrs/a/Leveraging-The-Cloud-For-Your-Dvr.htm

[9] Adam Thursby, “What Will Your Next DVR Look Like”, http://dvr.about.com/od/cabledvrs/a/What-Will-Your-Next-Dvr-Look-Like.htm

[10] Dr. Rao Mikkilineni and Vijay Sarathy, “Cloud Computing and the Lessons from the Past”, 2009 18th IEEE International Workshop

[11] Amazon Web Services, “Amazon Elastic Compute Cloud”, Service Highlights, http://aws.amazon.com/ec2/

[12] “Moving to the Cloud”, IEEE news source, Questions and Answers, http://theinstitute.ieee.org/opinions/question/moving-to-the-cloud873

[13] ETSI TS 102 034: "Digital Video Broadcasting (DVB); Transport of MPEG-2 TS Based DVB Services over IP Based Networks".

**First Author** Bharath Madhavan is with Cisco for the past 6 years and has a total industry experience of 16 years. In Cisco, he has worked in several projects related to MPEG technologies including DVR, VOD, EPG, PPV applications and several Conditional Access systems including Nagra and PowerKey technologies. He has been part of the Videoscape solution team, working on the client software effort focusing mainly on DRM and ABR technologies and as a team, has successfully integrated Playready and HLS technologies into our stack. Bharath has a Bachelors from IIT Madras and Masters from University of Maryland and another Masters from Johns Hopkins University. Prior to Cisco, Bharath has worked as an embedded engineer in Networking and Storage industries in the related areas of ATM, VoIP, SCSI, iSCSI, RAID, SAN and NAS technologies. After his Master’s degrees, Bharath worked in USA in various companies for about 10 years and then joined Cisco Inc, Atlanta in 2006. He then moved to India and joined the Cisco office in Chennai in January 2011.

**Second Author** Murugesan Nallathambi is with Cisco as part of the Architecture team and has in depth experience in Consumer electronics software development and has been working with Cisco for the last 6.5 years. He has contributed heavily to multiple programs including Open TV, Android, PowerKey, Videoscape & RDK projects. Currently, he splits time as a product owner for the base RDK and diagnostics team. He is actively involved in initiatives like WebRTC, Miracast, eMBMS & has published an IEEE paper in the wireless space.