SMART ABR: THE FUTURE OF MANAGED IP VIDEO SERVICES

John Ulm, Ajay Luthra, Praveen Moorthy, Mark Schmidt, Haifeng Xu
Arris, Inc. (formerly part of Motorola Home)

Abstract

This paper documents the research and lab results for using Adaptive Bit Rate (ABR) protocols for a managed video service. It quantifies the issues with unmanaged ABR including unfairness, instability and inefficiencies. It then explores some potential solutions including using either CMTS QoS or Server based algorithms in the cloud.

This ABR research has led to the evolution of Smart ABR (SABR) allowing operators to provide a first rate video service with exceptional Quality of Experience while retaining the underlying benefits of Adaptive protocols. The paper highlights lab results showing the optimization and Video Quality achieved. With SABR, operators can significantly increase their IP Video capacity while gracefully handling congestion and providing an improved user experience.

INTRODUCTION

Adaptive Bit Rate (ABR) protocols have become the mainstay of multi-screen devices like tablets, smart phones, gaming devices and Smart TVs for accessing Over-The-Top (OTT) video content. Because of their explosive popularity, it is highly desirable for an operator to provide existing video services to these devices. However, consumers will expect the same Quality of Experience (QoE) to which they are accustomed with today's primary TV screen delivery.

The ABR protocols have been optimized to operate over an erratic internet connection. However, the ABR client based control with no insights into system behaviors has demonstrated many inappropriate behaviors. The ABR client's greedy behavior leads to significant unfairness, instability and inefficiencies. These traits are not suitable for an operator to offer a true managed ABR video service with the associated QoE.

Operators have a number of challenges in offering a "managed" ABR video service. The limited bandwidth makes it challenging to support the demand of a large number of concurrent users while maintaining good video quality for each user. In addition, existing implementations of ABR client controlled distribution mechanisms are not very efficient. They tend to under utilize the available bandwidth and provide uneven visual qualities to the clients. Therefore, understanding the issues around delivery of ABR over the DOCSIS network will be crucial for MSO's video service delivery, and for their ongoing profitability.

Research into ABR has led to the evolution of Smart ABR (SABR). By adding some cloud based intelligence back into the system, the operator can regain control to provide a first rate video service with exceptional Quality of Experience while retaining the key underlying benefits of Adaptive protocols. SABR is a server controlled system that can manage the bit rates and video quality that each client receives.

Intelligence in the SABR server maintains client state, available client download bandwidth or channel capacity, and a measure of "reasonable" client video quality. "Reasonable" could be dependent on client attributes such as client display size or the type of video content being watched. Based on that intelligence, the SABR server controls what bit rate each client gets. This can prevent oscillations in network bandwidth utilization and increase network utilization.

The paper discusses lab results showing the optimization and Video Quality achieved. The SABR system is compared in detail to traditional unmanaged ABR delivery as well as a system with enhanced CMTS QoS. With SABR, operators can significantly increase their IP Video capacity while gracefully handling congestion and providing an improved user experience.

ABR Overview

Adaptive Bit Rate (ABR) is a delivery method for streaming video over IP. Adaptive streaming uses HTTP as the transport for small video chunks of approximately 2-10 seconds each. This enables the content to easily traverse firewalls, and the system scales exceptionally well as it leverages traditional HTTP caching mechanisms in the CDN.

Adaptive streaming was developed for video distribution over the Internet. In order to deal with the unpredictable performance characteristics typical of this environment, ABR includes the ability to switch at chunk boundaries between different encodings of the same content. This is illustrated in Figure 1. Depending upon available bandwidth, an ABR client can choose the optimum encoding to maximize the user experience.

The server stores several chunk sizes for each segment in time. The ABR client predicts the available bandwidth and requests the best chunk size using the appropriate URI. Since the ABR client is controlling when the content is requested, this is seen as a client-pull mechanism, compared to traditional streaming where the server pushes the content. Using URIs to create the playlist enables very simple client devices using web browser-type interfaces. A more in-depth discussion of ABR video delivery can be found in [Ulm].

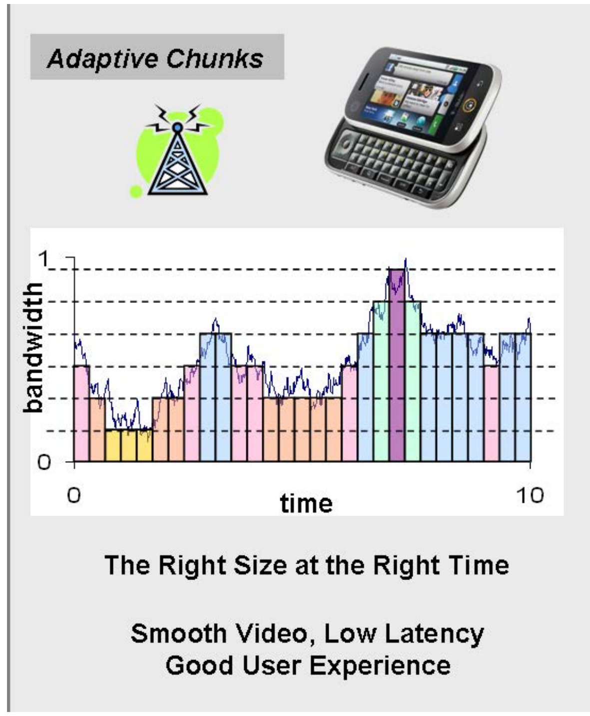


Figure 1 Adaptive Streaming Basics

Importance of ABR – 2nd and 3rd Screens

ABR based video streaming has become the de-facto standard for video delivery to IP devices such as PCs, tablets, smart-phones, gaming devices and Smart TVs. ABR clients are typically shipped with (or are available for download to) these devices as soon as they are released. Given the short lifetime of this class of device this is a key enabler, especially compared to the time required to deploy software to traditional cable STB devices. As mentioned previously, ABR delivery simply requires an HTTP connection with sufficient bandwidth so that it is available both on net and off net. With these advantages, essentially all video delivery to second and third screen devices uses this mechanism.

ABR vs. Current Managed Video Delivery

ABR video delivery has a number of very significant differences to both MPEG video delivery and streamed IP video delivered over Real-time Transport Protocol/User Datagram Protocol (RTP/UDP) as used in a Telco TV system. Foremost, ABR has been developed to operate autonomously over an unmanaged generic IP network.

- *The client device decides on bit rate (i.e. bandwidth) decisions based on* <u>*its*</u> *interpretation of network conditions.*

This is fundamentally different from the approaches used for existing MPEG or conventional streamed UDP video delivery, where devices under the direct control of the network operator make the important decisions relating to bandwidth. Thus, in MPEG delivery, the encoding, statistical multiplexing and streaming devices determine the bit rate for a given video stream. These devices are under control of the service provider. In contrast, the behavior of ABR clients is specified by the CPE developer which, in general, will be a third party outside the service provider's control.

An ABR client selects a file chunk with a bit rate that it believes to be most appropriate based on a number of factors including network congestion (as perceived by the client) and the depth of its playback buffer.

- *Thus the load presented to the network can fluctuate dramatically.*

Operators in a controlled network can guarantee that adding new user sessions do not impact existing users. Once resources are exhausted, any additional session requests will be denied, introducing a probability of blocking into the system.

ABR clients join and leave the network as users start and stop applications. From a network perspective, there is no concept of a session with reserved resources or admission control. Again this is the antithesis of MPEG or UDP video in which the control plane operates to request and reserve network resources and determines when to admit users. In a pure ABR model with network congestion, each new session will reduce the bandwidth available to all existing sessions rather than be denied.

- *Thus, all users may see a variation in video quality as other ABR clients start or change bit rates.*

With MPEG or UDP streaming video delivery, congestion control is not relevant as the control plane provides admission control to ensure it does not occur. When ABR is used for video delivery, congestion control is a potential issue. The situation is complex in that three levels of congestion control mechanisms are involved operating at different layers in the protocol stack. At the media access control (MAC) level, the CMTS is responsible for scheduling downstream DOCSIS traffic. Operating at the transport level is standard Transmission Control Protocol (TCP) flow control based on window sizes and ACKs. Finally, at the application level the client can select the video bit rate to request. The latter two levels of control (TCP and application) are the responsibility of the ABR clients and as such are outside the control of the network operator. Interaction between these flow control mechanisms is not well documented and may have unforeseen impacts.

In summary, ABR clients base their decisions on what to request based on their local knowledge and observed conditions rather than on an overall view of the network conditions. This is in contrast to MPEG or UDP streaming where the network operator provisions the video bit rates based on knowledge of the end-to-end network and expected loads.

Adaptive Bit Rate streaming is deployed today in a number of implementations, including MPEG Dynamic Adaptive Streaming over HTTP (DASH), Apple HTTP Live Streaming (HLS) and Microsoft HTTP Smooth Streaming (HSS). Due to the popularity of HLS and the abundance of iPad and iPod devices available for our testing, HLS was used in our tests. The techniques and solutions described in the paper may be applied to other ABR formats that rely on HTTP delivery of segmented content.

POTENTIAL MANAGED ABR SOLUTIONS

ABR Standards Investigations

The MPEG DASH Ad Hoc Group has been investigating the use of quality-driven streaming in a DASH environment. In this approach the client is provided with additional video quality information that the client can use during its segment selection and rate adaptation process. This info is generated during content preparation, converted to an estimated video quality measure and carried with each Media Segment.

While early experiments have shown improved QoE, MPEG DASH does not specify a normative client implementation or behavior, so the full benefit of client-directed, quality-driven streaming is dependent on well behaved clients using similar algorithms for segment selection/rate adaptation.

While this approach may improve QoE for a given client, it is not clear that this will solve the collective issues of unfairness and instability across a group of clients as seen in unmanaged ABR. Also, operators must offer a video service across a wide range of clients that must include others besides DASH. For these reasons, DASH was not considered as a managed ABR solution.

CMTS as Control Point

For users on an HFC network, IP traffic will always flow through the same CMTS port to reach a user at home. As the shared CMTS to CM link is normally the “narrow pipe” in the video distribution network, this is where congestion would be expected. Therefore the CMTS can potentially provide a useful control point to manage ABR traffic.

The DOCSIS standard provides very complete Quality of Service (QoS) functionality which may be useful for managing ABR traffic. If a packet matches an installed classifier it will be mapped to a specific Service Flow and then forwarded based on the parameters associated with that Service Flow. Classification is based on matching fields in the packet header such as Source &/or Destination IP address, port & type and Differentiated Services Code Point (DSCP) fields.

Therefore it is possible to recognize a managed ABR video packet stream from a well known source address (e.g. video server) or IP subnet. The CMTS could then provide preferential QoS treatment for the operator’s managed video flows. One of the goals of our research was to verify the impact of using CMTS QoS for managed ABR video service.

The DOCSIS infrastructure has a mechanism to dynamically setup and control Service Flows based on the PacketCable™ Multimedia specification [PCMM]. This provides a potential mechanism to implement resource reservation at the session level. It requires a session establishment and teardown mechanism. However, this may be problematic with the distributed nature of adaptive streaming protocols and may have scaling issues. The control plane topic is outside the scope of this paper.

Cloud based Server Solutions

In conventional ABR video distribution, the ABR client determines the bit rate of the next file to download from the options in the playlist and retrieves this directly from the Content Delivery Network (CDN). By adding intelligence into the cloud, this decision could potentially be overridden from the network in a number of ways. This is referred to as Smart ABR or SABR.

The playlist file provides the bit rate options specified by the service provider. Normally this selection would be statically provisioned and implemented by the encoding and packaging processes as the video asset was processed. It is conceivable that the playlist may be manipulated and the network can regain control of what bit rates are available to the ABR client.

During peak utilization, existing managed video delivery uses admission control to block new users from accessing the system. With SABR, an operator could gracefully handle congestion during peak times with no blocking of users, but rather a slight degradation of video quality. This reduction in quality during peak times is analogous to statistical multiplexing in legacy MPEG video. During peak times, the statmux reduces bit rates across the various video streams to fit within its channel. The SABR system has an advantage in that it will be multiplexing over a larger channel using DOCSIS bonding.

In a SABR system, all the clients are controlled from the server side. The system level intelligence in the server understands the state for every client; the available bandwidth for each client; and a "reasonable" visual quality of the video for a given size of display and attributes of video etc. Based on that intelligence the server controls what bit rate each client gets. This avoids the oscillations and increases network utilization.

ABR TEST METHODOLOGY

Our goals were to research ABR behaviors in a working environment. First item was to replicate and quantify existing unmanaged ABR characteristics. Then test these same conditions for a CMTS QoS based solution and a cloud based SABR Server solution.

To analyze our lab results, it is important to understand some of the fundamental operation of the ABR client.

ABR Client Characterization

The ABR client plays a critical role in the operation of adaptive protocols. For an operator trying to provide a differentiated quality of experience, it is important to understand how different ABR clients behave under various circumstances.

Previous work [Cloonan] discussed results from a simulator. Our goal was to capture live client interaction. Operation during steady state was relatively stable.

HLS Client Model

Because of the abundance of iPad and iPod devices available for our testing, HLS devices are used in these tests. A simplified HLS client diagram is illustrated in Fig 2.

A stored HLS program such as Video On Demand (VOD) assets have a manifest that lists all of the program's available media chunks or segments and the player downloads chunks starting from the earliest. When the client plays a stored program, it first reads a manifest file (playlist) from HLS server with specified URI, parses the content of the file, and starts to request HLS chunks sequentially starting with the lowest sequence number.

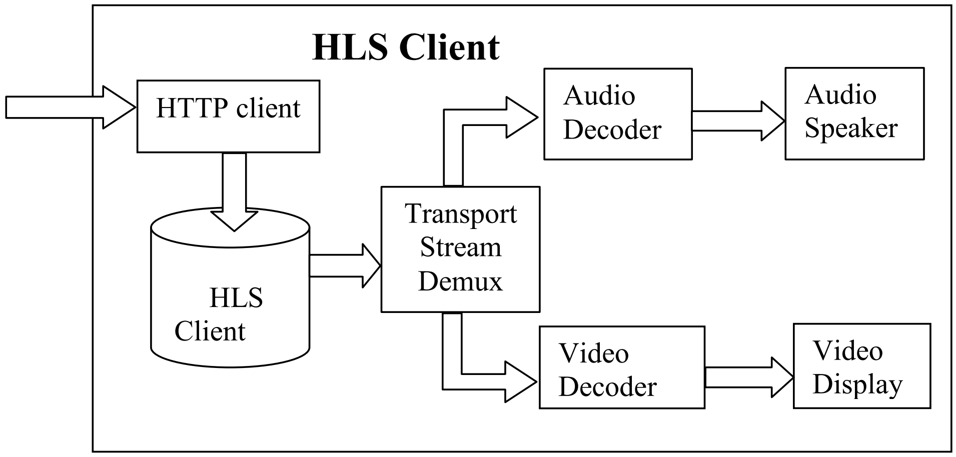


Figure 2 Diagram of HLS Client

A live Linear HLS program's manifest changes as new content are created; a sliding-window of chunks is given to the player and the player may or may not download the earliest chunk in this manifest.

The video play back does not start until the buffer hits a certain threshold. Therefore, if a client is not able to fill its buffer fast enough to a certain designed level, it will take longer to start the play back. Once the client fills up its buffer, it moves into the Playback stage.

In the Playback stage, the client fetches one HLS chunk during each chunk period. In other words, steady state is achieved and the overall download chunk speed matches its real-time play speed.

The two phases are illustrated in Figure 3. As can be seen, a client puts higher stress on the network bandwidth during the Buffering stage than that in the Playback stage as the clients try to buffer multiple segments as fast as they can. This introduces the following inefficiencies:

- *The HLS client buffer is necessary to deal with network jitter and varying bandwidth. Therefore, it requires more overall bandwidth during Buffering stage to provide this cushion.*
- *The HLS client relies on a combination of TCP/IP mechanisms at the low layer and adjusting video bit rates at application layer to deal with network variations. To provide the visual quality that does not vary too fast, a HLS client will not utilize full network capacity.*

As a result of these, we have observed clients in our labs that may leave up to 50% of the available network bandwidth unutilized Note: this number will vary depending on specific HLS client implementation.

During this startup period, the clients are also calculating the available bandwidth and may decide to switch bit rate. This action may cause some segments to be re-fetched with the new resolution. Overall, the differences between clients seemed fairly subtle for startup.

Once HLS clients retrieving a VOD playlist content reaches steady state Playback stage, it may have 50-60 seconds worth of content in its buffer. Live content tends to have a limited playlist available to the client, preventing large buffer build up.

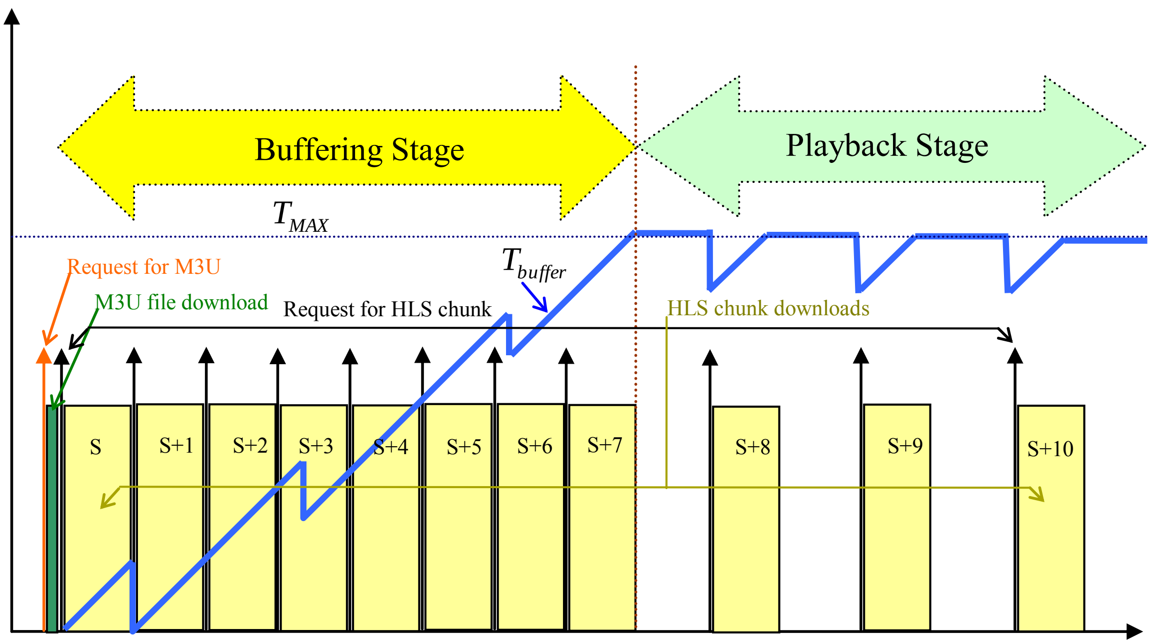


Figure 3 Two phase client stages

During playback, if an ABR client detects a significant enough decrease in available network bandwidth, it decides to switch to a lower available bit rate. This causes the client to move back to the Buffering stage and the client often downloads multiple chunks of the lower rate stream that correspond to chunks it has already downloaded from the higher rate stream. This could allow the client to seamlessly stitch video and/or audio in its decoder buffers. This download overlap may be as little as 2-4 seconds or as much as 12 seconds but still represents increased network load.

Conversely, when a client detects that its available download bandwidth has increased sufficiently, it switches to a higher bit rate and re-loads chunks corresponding to already downloaded media of the lower bit rate. The client is once again in the Buffering stage. However, these downloads can be substantial and clients have been observed to re-download enough segments to refill part or even the entire 50-60 sec buffer described above. Also, the client may quickly ratchet through multiple video rates as it tries to determine the optimum rate. These behaviors are significant contributors to the bandwidth oscillations among ABR clients.

Multiple Client Interactions

Based on the above characterization, operators must be aware of some potential problems. As was discussed, there is a burst of additional traffic during startup and when switching bit rates. For managed video delivery, the overall system must be capable of handling this additional traffic burst.

Actively managing ABR video traffic may be challenging given that every ABR client may be operating its own disjoint algorithm. This is also compounded since client behavior may change with the download of an updated revision. Bandwidth stability may become a concern if multiple clients become synchronized.

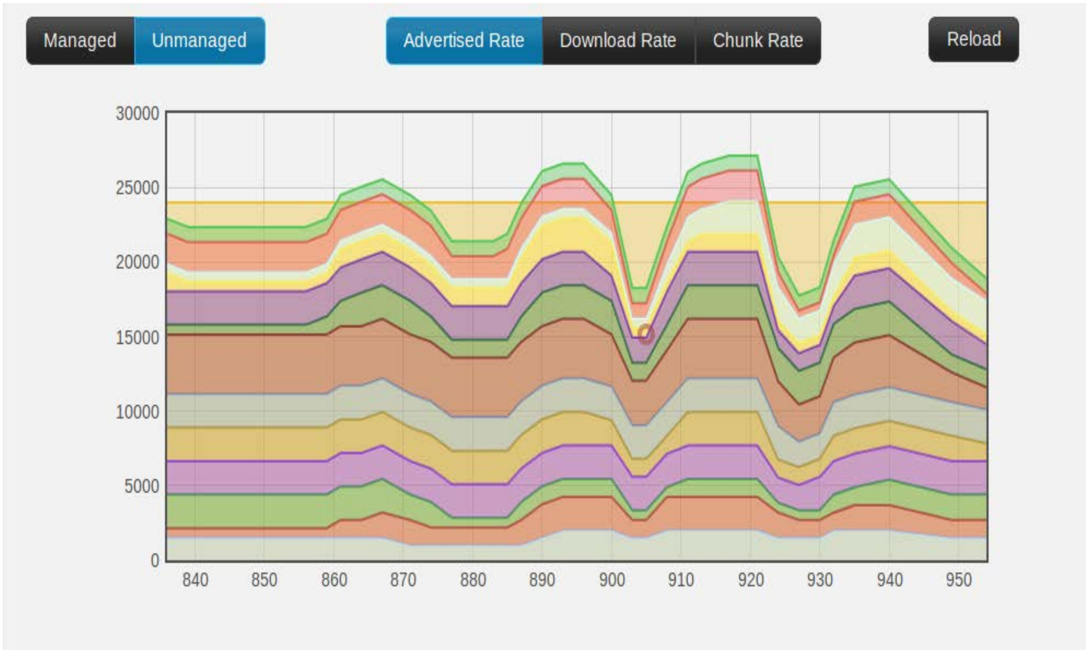


Figure 4 Multiple Client ABR Oscillation Example

For example, the network becomes congested causing a group of clients to lower bit rates. If these clients then sense that bandwidth is available (i.e. it is released due to downshifting by other clients), there may be a surge in traffic that causes congestion, and the cycle repeats. This oscillatory behavior is demonstrated in Figure 4 which shows the ABR bit rates versus time of thirteen HLS clients in a stacked bandwidth plot. The frequent stream bit rate changes results in changing video quality at the clients as well as inefficient network utilization. These issues of stability and fairness have been discussed in other papers such as [White] and [Adams].

ABR TEST SETUP

Our lab test setup needed to allow tests to replicate and quantify the issues with unmanaged ABR delivery and then show the impact of managed ABR solutions. A block diagram of the ABR test setup is shown in Figure 5. The test setup was explicitly designed to show a number of variations. More details on the lab test setup can be found in the appendix.

In these tests, there were only HLS video streams present and no other background Web data or VoIP traffic in the system.

Network Configuration

The overall network configuration consisted of an HLS Server, a CMTS, a half dozen cable modems with WiFi and then ten iPads/iPods. Since a primary goal was to observe behavior under network congestion, the CMTS was configured to a single downstream channel operating at 64-QAM. This provides approximately 25Mbps of user bandwidth. The Upstream channel was configured to operate at 10Mbps, sufficiently large that it shouldn't impact test results by delaying TCP ACKs.

A HTTP server hosted multiple ABR VOD media streams and was connected behind the CMTS along with a PC server running DHCP and TFTP servers used for cable modem configuration.

The six cable modems were dual-band 802.11n WiFi capable. Four cable modems used the less congested 5 GHz WiFi bands while iPods were associated with cable modems on the 2 GHz WiFi band.

ABR Video Clients

The test setup consisted of 7 iPads and 3 iPods clients. The number of clients per cable modem was intentionally varied from 1 to 3 to measure the impact of having multiple streams per home.

Apple iPads of various generations and HLS client software, iOS, was selected to observe if there were any impacts from different generations of iOS versions.

The clients were started in sequential order as listed in Table 1 in the Appendix. No attempt was made to allow each client to complete its buffering phase so it was expected there would be high offered load as the client's began their downloads one after the other. The tests lasted at least 20 minutes during which time logs were maintained of the advertised and actual bit rates of the HLS media segments (chunks) retrieved by the clients, the associated times of those chunk requests, the downloaded chunk sequence number, and the video quality.

Video Content

The HLS video streams varied in content complexity and resolution. The tests include both VGA (480p30) and HD (720p30) resolutions. Video clips were chosen such that a metric (i.e. PSNR data) was available for measuring video quality. A detailed description of how PSNR (Peak Signal to Noise Ratio) is calculated is in the Appendix.

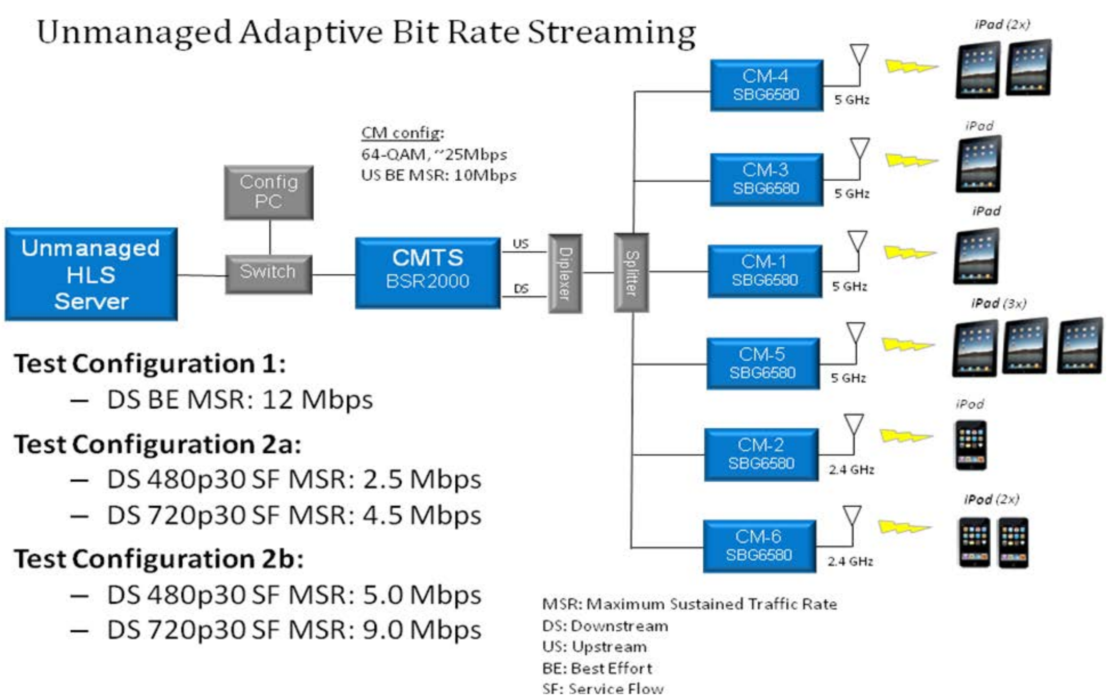


Figure 5 Adaptive Bit Rate Streaming Test Setup

The 480p30 VGA content supported four bit rates at 650kbps, 1.2Mbps, 1.7Mbps and 2.2 Mbps. The 720p30 HD content supported four bit rates at 1.0, 2.0, 3.0 and 4.0 Mbps.

CMTS QoS

For the unmanaged ABR tests and the SABR Server tests, the cable modems were configured for simple best effort traffic with a burst rate of 12Mbps. This rate was chosen to be less than the channel capacity while sufficiently large to support multiple video streams per cable modem.

For the CMTS QoS tests, each video client had its own Service Flow statically configured. One test was run with the max burst rate slightly higher (i.e. ~12%) than the HLS client's max video rate, then the remaining CMTS QoS tests were run with the max burst rate set to twice the max video rate (i.e. 100% extra Burst capacity).

UNMANAGED ABR TEST RESULTS – DEALING WITH A DRUNKEN TEENAGER

While analyzing the unmanaged ABR test results in the lab, an observation was made that the ABR clients were like dealing with a drunken teenager. Both are very erratic, self centered, greedy, unpredictable and hard to manage. This causes a slew of problems that are detailed below. The most intriguing observations occurred during startup and when video bit rates were forced to change.

The results of one of our unmanaged ABR tests are shown in Figures 6a, 6b & 6c. Figure 6a shows the aggregate bit rate from all 10 video clients for both the bit rate selected and the actual bandwidth consumed. The difference between these two lines is the result of clients being in the Buffering stage and requesting additional chunks to fill its buffers. Figure 6b shows a group of four charts depicting the instantaneous bit rate that each of the 10 HLS clients are requesting. The first, third and fourth charts show clients grouped by the cable modem that they share. The second chart is a collection of the three video clients with their own cable modem. Fig 6c is a collection of four charts showing the estimated video buffer depth for each of the 10 HLS clients. The four charts are organized in the same manner as in Fig 6b. [Note – this same layout is then used for the CMTS QoS results (Fig 7a-c, 8a-c) and the SABR results (Fig 9a-c).

Network Utilization

Our first observation is the network utilization from Fig 6a. The aggregate bandwidth consumed is very close to the 25Mbps capacity of this DOCSIS downstream. The average bandwidth consumed after that initial startup is ~23.6Mbps. The second line in Fig 6a shows the aggregate of all current bit rates selected. This averages about 19.2Mbps after the initial startup. This means that the unmanaged ABR scenario utilizes ~77% of the total channel capacity. The ABR clients have improved over the years. Early generation software was only seeing ~50% network utilization.

Instability

The next observation is on the stability of the system. In looking at the requested bit rates in Fig 6b, half of the video clients appear fairly stable after a couple minute startup period while the other half are oscillating for the entire test. These 5 HLS clients averaged between 1 and 1.5 bit rate changes per minute after the startup period.

Looking at Fig 6c gives an insight into the impact on this instability. The 5 volatile HLS clients had trouble maintaining their video buffer depth. This volatility would have been disastrous for live linear content.

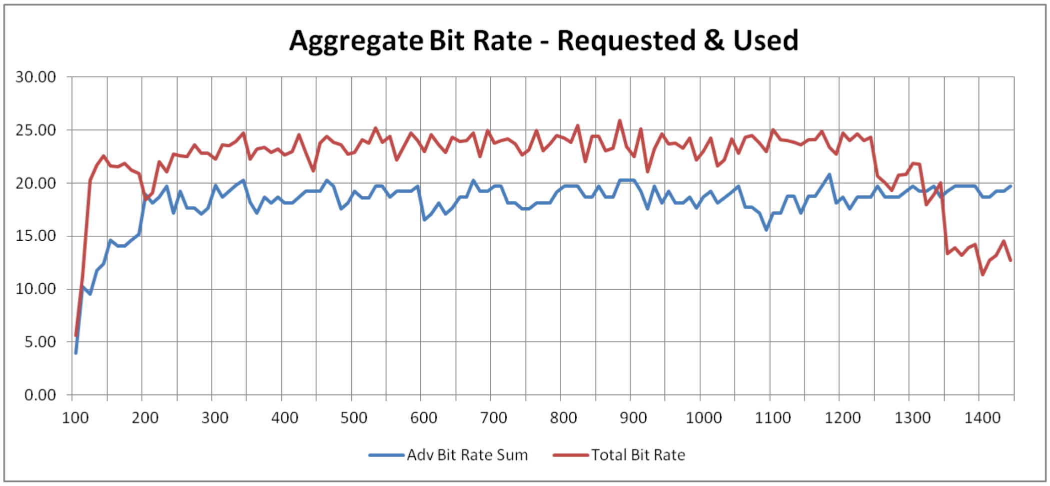


Figure 6a Unmanaged ABR – Aggregate Rates

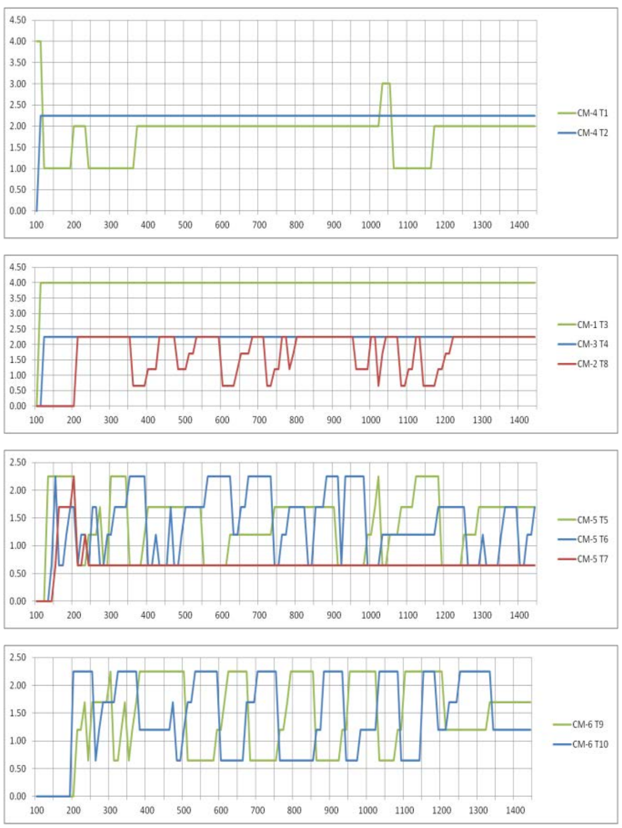


Figure 6b Unmanaged ABR – Bit Rates

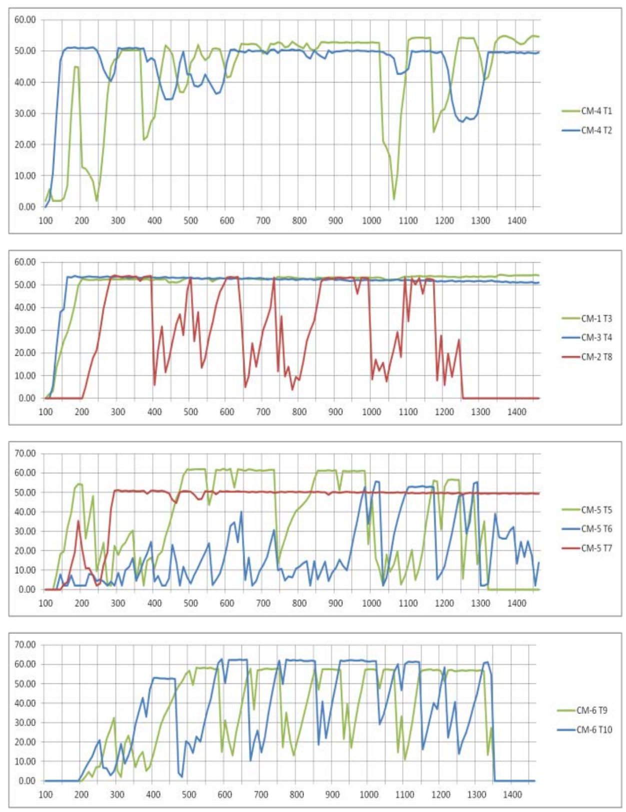


Fig 6c Unmanaged ABR – Buffer Depth

Unfairness

A key attribute for managed video service is maintaining fairness across clients. In running multiple unmanaged ABR tests, we observed multiple types of unfairness.

While both HD clients appear stable [see Fig 6b, CM-4 T1, CM-1 T3], one settles in at its maximum bit rate of 4Mbps while the other sits at 2Mbps most of the time. Similarly, two stable VGA clients settled at its max rate of 2.2Mbps (i.e. higher than one of the HD clients!!) and a third stable VGA settled at its min rate of 650Kbps.

To add insult to injury, one of the VGA clients at 2.2Mbps was a low complexity News clip (i.e. talking heads) while the HD clip sitting at 2Mbps was a high action and complexity Football sequence.

Other forms of unfairness may be introduced when network congestion causes video bit rate changes. Some clients may decide to change while others remain at current bit rates, resulting in disparity between clients.

It was also observed that in general the modems with multiple HLS clients struggled more than modems with a single client. The CMTS is trying to distribute bandwidth fairly among cable modems, so a home with three HLS clients find themselves competing for that home's bandwidth while another client may have the home's bandwidth all to itself.

We also observed that some HLS clients with older software struggle more compared to the clients with newer software. While overall utilization has improved over the years, this has apparently been done while making newer clients more aggressive and taking advantage of clients with older software.

In summary, we observed fairness issues with each of the following:

- *Client vs. client*
  - *Screen size (HD vs. VGA)*
  - *Video Complexity (Low vs. High)*
- *HLS client versions*
  - *newer more aggressive*
- *Multiple clients per home*
  - *WiFi impacts as well*

Video Quality

Some qualitative observations were made on perceived video quality while running the tests. The clients with stability issues would experience frequent changes in bit rates that resulted in reduced viewer experience. The changes in resolution were obvious. Some clients were oscillating from the max rate to their min rate. It appears that it would be better QoE if the rate stayed constant at a lower value then continually adjusting between different rates.

Even for the stable clients, the poorer resolution was noted for the HD client stuck at the lower rate and the VGA client stuck at its lowest rate.

Perhaps the worst impact on QoE was the buffer under-runs that cause the screens to freeze. Looking at Fig 6c shows how a number of clients struggled to keep an adequate amount of video in its video buffer. Two clients had aggregate buffer under-runs of 5-10 seconds and one client had over a minute of time with buffer under-runs.

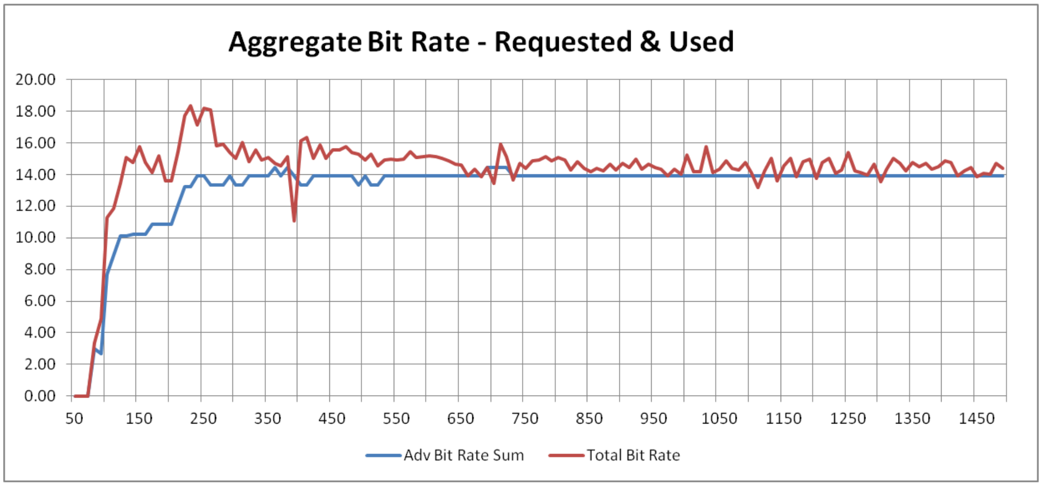
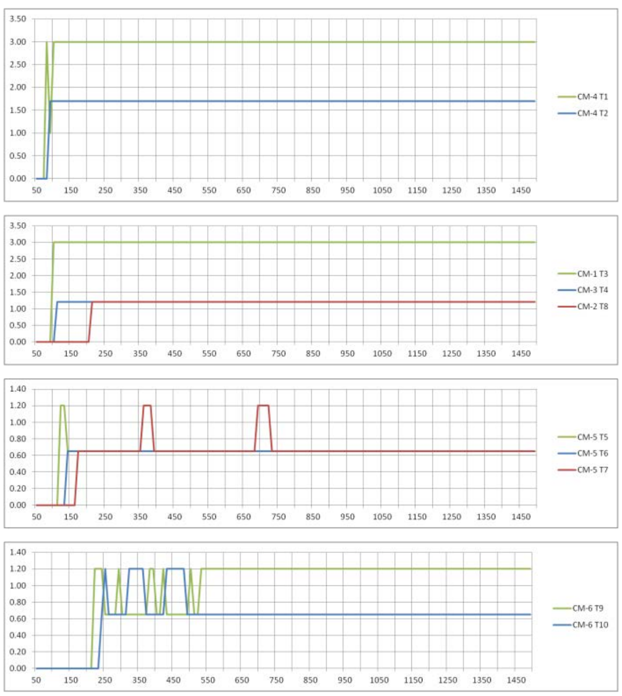


Figure 7a CMTS QoS, 12% Burst – Aggregate Rates



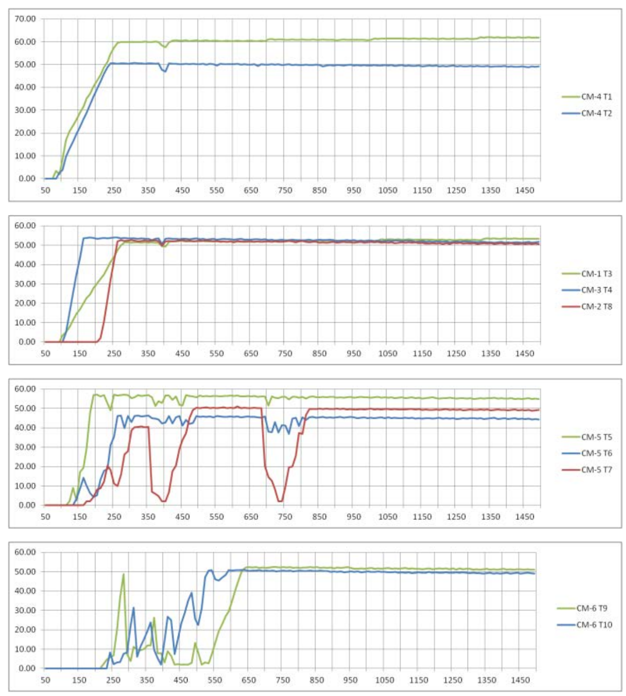


Fig 7b CMTS QoS, 12% Burst – Bit Rates

Fig 7c CMTS QoS 12% Burst – Buffer

CMTS QOS TEST RESULTS – CAVALRY TO THE RESCUE??

Before the CMTS QoS tests were even run, there were some high expectations that the CMTS could fix most all of the issues seen above for unmanaged ABR. With a dedicated Service Flow for every HLS client, it seemed like all clients would get their fair share of bandwidth thanks to the Weighted Fair Queuing scheduler in the downstream. With separate weighting on each Service Flow, even the HD clients should get a proportionately larger bandwidth share.

CMTS QoS Tests with 12% Burst Overhead

For the first CMTS QoS test, we intentionally kept the Service Flow Max Service Rates (MSR) just above the maximum video rate being offered. So VGA clients got a 2.5Mbps MSR for a 2.2Mbps max video rate and the HD clients got a 4.5Mbps MSR for a max 4Mbps video rate. The total aggregate MSR for the 10 Service Flows was 29Mbps, so it should come close to filling the channel. The results for this test are shown in Fig 7a-c.

Surprisingly, the aggregate bandwidth used peaked around 15Mbps and the average of the requested bit rates was ~14Mbps. So, this configuration had a network utilization of only 56%. This clearly shows that the HLS clients need a more substantive burst capability while in buffering mode. On the positive side, because the network was not congested, the system was extremely stable as can be viewed by Fig 7b & 7c.

Taking a closer look, both HD clients settle in quite comfortably at 3Mbps video rates with a 4.5Mbps MSR. For the VGA clients, there is a range of video rates from 4 @ 650Kbps, 3 @ 1.2Mbps and one @ 1.7Mbps. So, the CMTS scheduler was not able to correct all of the unfairness issues between the different VGA clients. Note that all but one of the clients settled at a bit rate that is less than half of its 2.5Mbps MSR.

The fact that network utilization was so low reinforces the complexity of the system with layers of flow control including TCP and the ABR clients. This area definitely deserves more research.

Note – video buffers in Figure 7c are generally quite stable (after startup) which bodes well for live Linear TV usage.

CMTS QoS Tests with Larger Burst Rates

The MSR was then opened up on the next set of CMTS QoS tests. The HD clients got a 9Mbps MSR and the VGA clients got a 4.5Mbps MSR, both slightly more than twice the max offered video rate. The key question now is the CMTS capable of fully utilizing the downstream channel while keeping the stability of the previous test. The results are shown in Fig 8a-c.

The network utilization does rise and appears to be close to the unmanaged ABR tests. The average bandwidth consumed after that initial startup is a tad less at ~23.1Mbps while the aggregate of all current bit rates selected averages slightly higher around 19.6Mbps after the initial startup. This bumps the network utilization up a hair to almost 80% of channel capacity.

Checking the stability of bit rates on Fig 8b shows that the two HD clients quickly snagged their max 4Mbps bit rate and kept it. Two of the early VGA starters locked onto a 2.2Mbps rate, but the remaining VGA clients did not fare so well. So, once a client can establish sufficient bandwidth demand, the CMTS QoS allows it to keep it. However, the remaining clients struggle with the remnants and show a lot of instability. This instability is apparent in the buffer depths of these clients in Fig 8c.

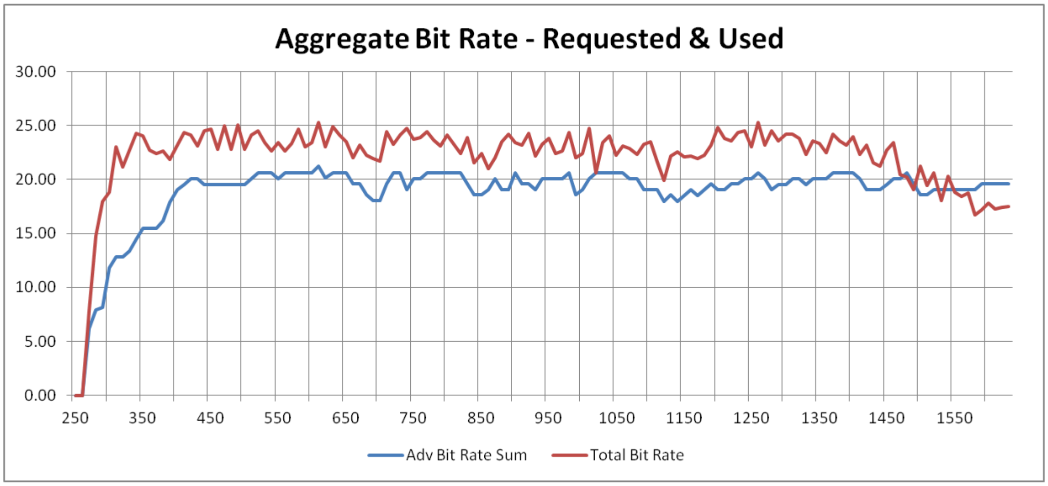
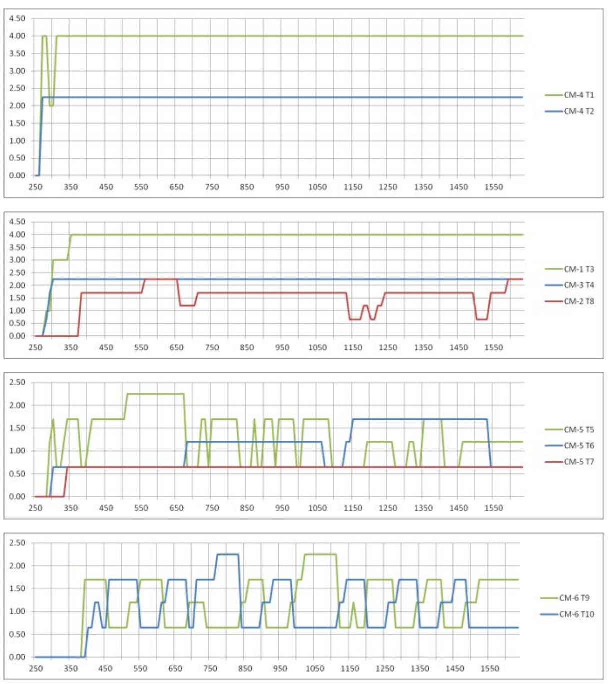


Figure 8a CMTS QoS, 100% Burst – Aggregate Rates



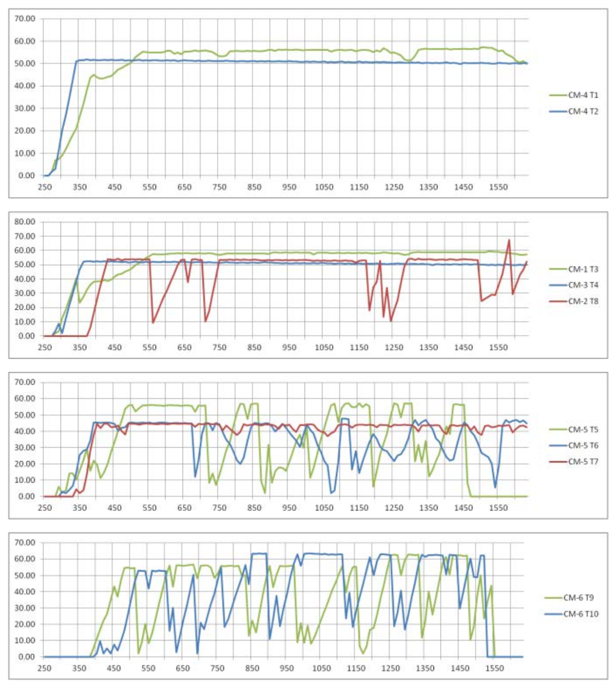


Fig 8b CMTS QoS, 100% Burst – Bit Rate

Fig 8c CMTS QoS, 100% Burst – Buffer

Another interesting observation is on the bottom chart for the two HLS clients sharing CM-6. These two clients are oscillating 180 degrees out of phase with each other. Most of the time the bit rates are bouncing between 650Kbps and 1.7Mbps. When increasing bit rates, there is a brief transition thru 1.2Mbps which increases the buffering overhead even more. It was noted that this cable modem was operating in the 2.4MHz WiFi band with multiple clients. This is another area for further investigation.

SABR SERVER TEST RESULTS – ORDER IS RESTORED TO UNIVERSE

In the previous CMTS QoS test configurations, the CMTS was trying to shape bandwidth to coax the clients into the optimal video rate selection. As was seen, the "drunken teenager" doesn't always cooperate. With the SABR Server testing, the bit rate choice is removed from the client and given to intelligence in the cloud. These results are shown in Fig 9a-c below.

The network utilization shows a definite improvement over the unmanaged ABR and CMTS QoS tests. While the average bandwidth consumed after that initial startup period is lower at ~21.8Mbps; the average video bit rate requested is ~10% higher at 21.1Mbps after the initial startup. This bumps the network utilization up over 84% of channel capacity. The SABR algorithms are still in its infancy, so there is a good opportunity to push the network utilization above 90% over time.

A first glance at the bit rates in Fig 9b might indicate significant instability. But in reality, the opposite is occurring as can be seen by the video buffers in Fig 9c. What is happening is that the SABR Server is intentionally adjusting the bit rates to try and maintain a constant video quality.

This is quite analogous to a Variable Bit Rate stream. The SABR Server must also trade off bandwidth needs between the different clients, so it performs the equivalent functions that a Statmux does in the MPEG world, only at a fraction of the computational complexity.

The net result of all of this is that SABR system produces a rock solid stable environment with excellent video quality and QoE. With the excellent control of the video buffer, this is ideally suited to distributing live Linear TV over ABR as well.

VIDEO QUALITY – THE REAL LITMUS TEST

As was seen in the previous unmanaged ABR and CMTS QoS tests, the steady state video bit rate was used to imply video quality. This metric no longer held with the SABR Server test configurations.

To do a more comprehensive study, our lab tests also measured the PSNR of every chunk delivered to each client. This allows a quantitative analysis of the delivered video quality. The appendix contains more details on how the PSNR is calculated.

Our analysis focus on two key aspects:

- *Percentage of chunks below 35dB*
- *Percentage of chunks above 40-45dB*

From our vast experience in video quality analytics, video below a PSNR of 35dB often translates to poor subjective video quality obvious to a common observer. As the PSNR improves much above 40dB, then the subjective video quality is getting past the point where a common observer might see any difference. In other words, having a PSNR that's too high means you are throwing bits away that a consumer will never observe.

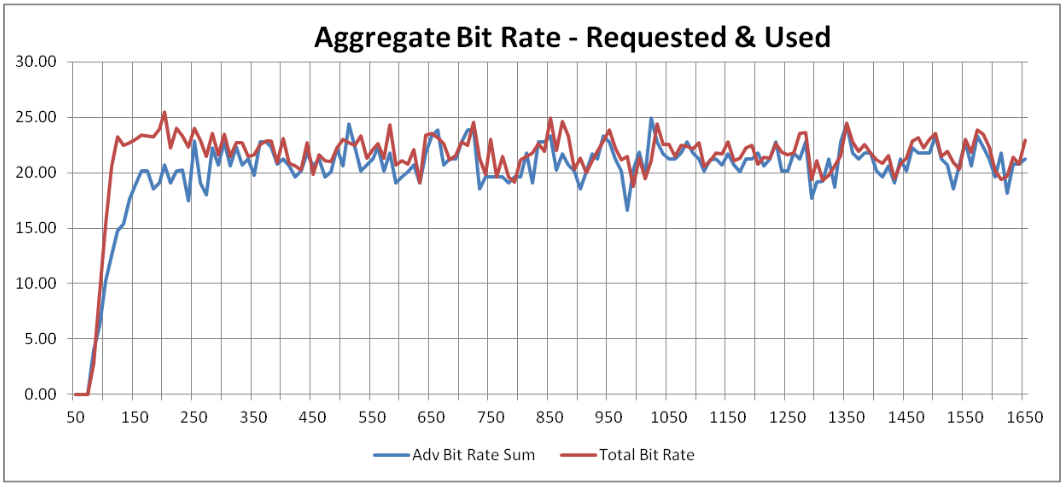


Figure 9a SABR Server – Aggregate Rates

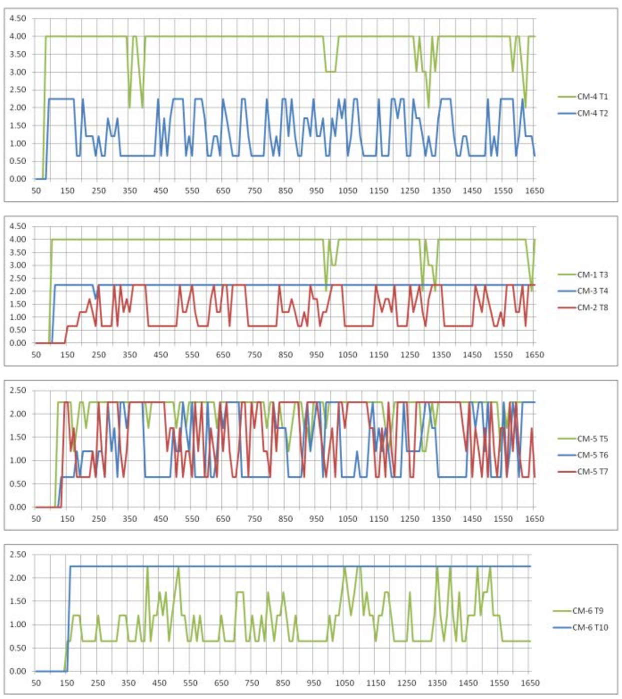


Figure 9b SABR Server – Bit Rates

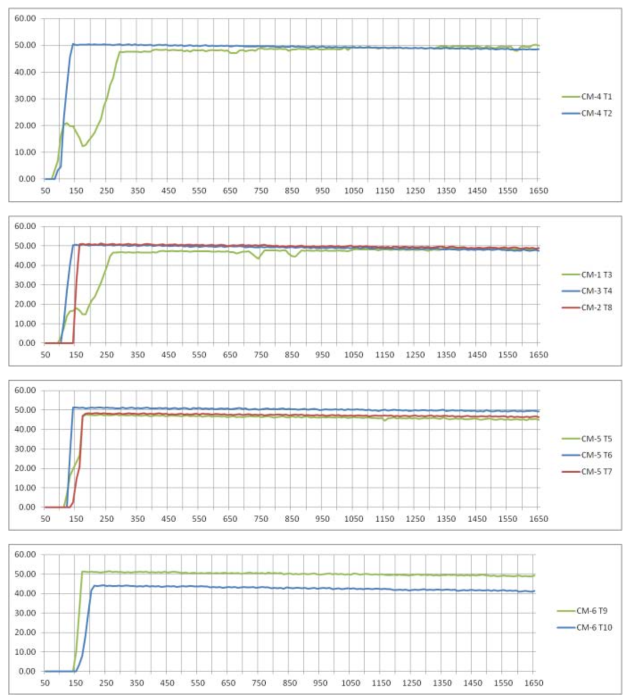


Figure 9c SABR Server – Buffer Depth

The results of the PSNR calculations for all the previous tests are shown in Fig 10a-e. A separate PSNR curve is drawn for each and every client. These are a Cumulative Distribution Function (CDF) that shows the % of chunks on Y axis that match a given PSNR on X axis. Given these desired thresholds, the ideal curve would be a virtual step function that stays at 0% until 35dB and then reaches 100% by 40dB.

Figure 10a shows the PSNR results for the unmanaged ABR test. Five of the HLS clients have a significant portion (i.e. 25%-50%) of their chunks below the 35dB threshold. With no surprise, these map to the clients with unstable bit rates. The other five stable clients had 30%-70% of their chunks above 45dB, so lots of potential video bandwidth is being wasted. Also notice the spread (i.e. width) of the curves. This illustrates the unfairness in video quality between the different clients.

The results for the first CMTS QoS test with limited 12% burst rate are shown in Fig 10b. Remember that this test was very stable including the video buffer depth, but settled at a lower aggregate bit rate. However, the PSNR results were only marginally better than the unmanaged ABR case. Three clients had about 30% below 35dB PSNR while one horrible client had ~70% below 35dB. The others did extremely well, but in fact too good. Four clients had 70%-80% above 45dB, so lots of video bits going to waste. The PSNR spread between the clients is comparable to the unmanaged ABR test as different clients settled at different bit rates.

The next CMTS QoS test with full burst rate capability is shown in Fig 10c. The PSNR results of the 8 VGA clients are almost identical to the previous CMTS QoS test. The only notable PSNR difference was the improvement for the two HD clients. This is not surprising as both clients went from mostly a 3Mbps video stream to a 4Mbps video stream between these two tests.

Now let's take a look at the SABR Server PSNR results in Fig 10d. It was shown earlier that the video buffers were extremely stable while the video rates varied quite a bit. As can be seen, it produces significantly better PSNR results than both unmanaged ABR and CMTS QoS tests. Most of the clients have negligible >2% of chunks below 35dB, with one client at 12% and two clients at 18%. A closer look reveals that most of these chunks were actually within 1dB of the threshold. If a 34dB threshold is used, then seven clients drop virtually to zero, two clients at 2% and one client at 6%.

There are also significantly fewer chunks above 45dB with six clients at ~90% and the rest at ~80% by 45dB. The spread between the clients is dramatically reduced as well, with the spread between the best and worst being reduced by a factor of three. This is the ultimate litmus test that SABR provides video quality fairness across all clients.

The final figure 10e shows a previously unmentioned test configuration. This test limited the SABR Server to a max output of 15Mbps. The bit rate and video buffer charts were almost identical to the other SABR test, just scaled to a lower rate. However, the PSNR results for this test are very interesting. Because there is less total bandwidth, all video clients were given proportionately lower video rates; which in turn impacts the PSNR value.

Four of the clients had 30%-50% below 35dB, with the remaining clients less than 20% below 35dB. On the low end, this is very close to the results produced by the unmanaged ABR test. Put another way, SABR delivered comparable video quality in almost half the amount of bandwidth (i.e. 13.5Mbps vs. 23.6Mbps) while providing fairness and stability.

Another interesting comparison is with the first CMTS QoS test in Fig 10b. Both of these operated with just under 15Mbps of bandwidth so it is close to an apples to apples comparison. The SABR results show a significantly narrower PSNR spread of ~5-7dB while the CMTS QoS spread hovers around 15dB. This indicates SABR provides significantly better fairness between clients then the CMTS QoS approach.

These results show the real potential of SABR to increase the video capacity for operators while maintaining a very consistent video quality across _all_ of its consumers.

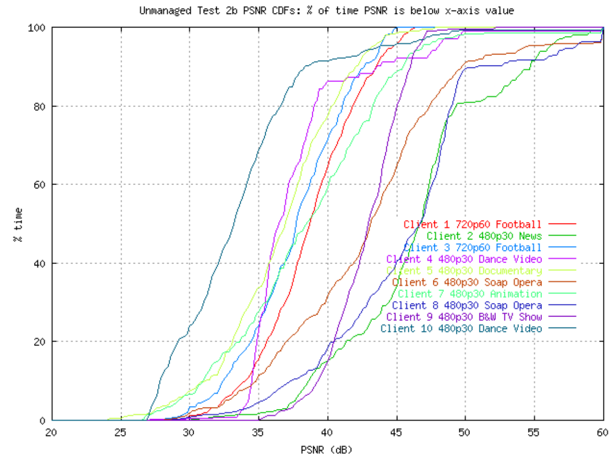


Figure 10c PSNR CMTS QoS, 100% Burst

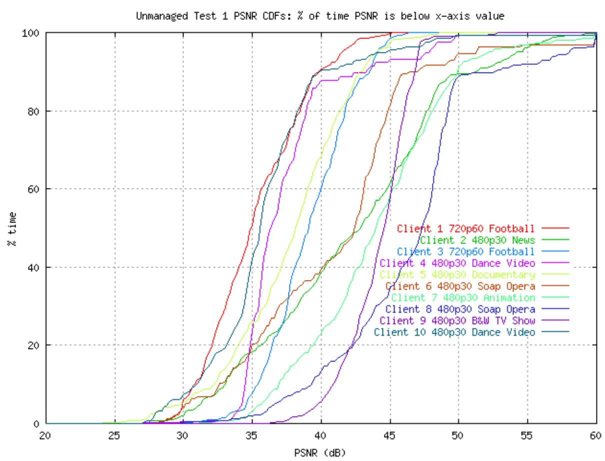


Figure 10a PSNR for Unmanaged ABR

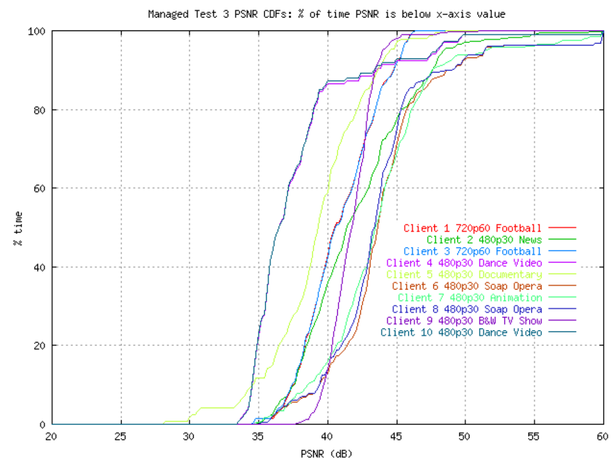


Figure 10d PSNR for SABR Server

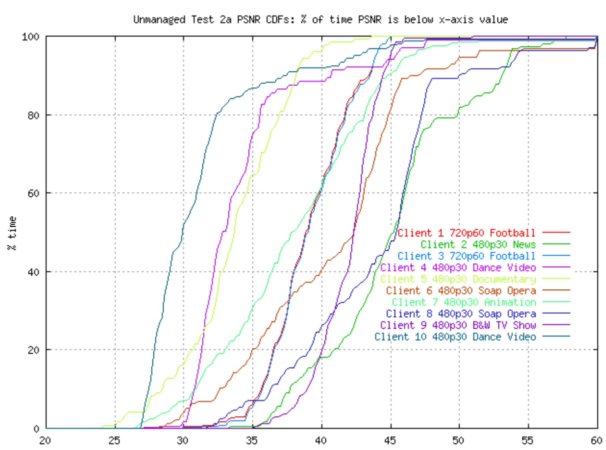


Figure 10b PSNR CMTS QoS, 12% Burst

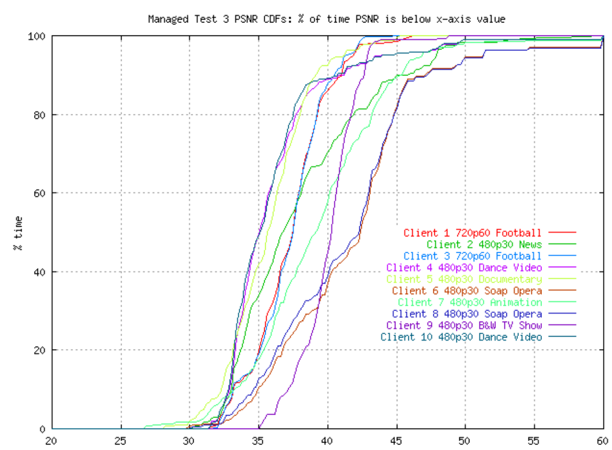


Figure 10e PSNR SABR Svr 15Mbps Cap

CONCLUSION

Operators have a number of challenges in offering a “managed” ABR video service. The limited bandwidth makes it challenging to support the demand of a large number of concurrent users while maintaining good video quality for each user. In addition, existing ABR client controlled distribution mechanism creates issues with QoE, network utilization, fairness and system stability. Our lab results presented confirms all of these conditions. In fact, fairness was an issue in multiple areas such as screen resolutions (HD vs. VGA), client software revision, shared home bandwidth and even the timing of when each client starts. While some clients were stable, others oscillated between bit rates and suffered video buffer under-runs.

A second series of tests were run leveraging enhanced CMTS QoS to try and fix the above issues. The results show only marginal improvements over the unmanaged ABR tests and the underlying issues of unfairness and instability still exists. Even had this approach fixed the problems, there are still concerns that the control plane could scale to support this solution.

Finally, our tests focused on a cloud based approach called Smart ABR (SABR) where the operator can regain control to provide a first rate video service with exceptional Quality of Experience while retaining the underlying benefits of Adaptive protocols. Our lab results show that all major shortcomings of unmanaged ABR can be addressed. The system becomes stable, network utilization improves and video quality fairness is provided across the entire client population while gracefully handling congestion and providing an improved user experience. Further tests suggest that SABR delivered comparable video quality in almost half the amount of bandwidth while providing fairness and stability

This lab testing is just the tip of the iceberg. Further research is warranted in areas such as optimizing SABR algorithms; mixing data traffic with the ABR video; investigating the impacts of WiFi in the home; further research into other CMTS QoS mechanisms; scaling testing to a much larger client population and understanding some of the subtle anomalies that appeared during our initial tests. It is important that the industry grasps the system dynamics for adaptive protocols.

With all of the promise seen with this initial testing, SABR should quickly become the future of IP Video.

ACKNOWLEDGEMENTS

In addition to our co-authors, I would also like to acknowledge Gary Hughes for his input and guidance into both the paper and our general research into adaptive protocols.

REFERENCES

[ADAMS] – “Optimizing Fairness of HTTP Adaptive Streaming in Cable Networks”; M. Adams, C. Philips, The Cable Show NCTA/SCTE Technical Sessions spring 2012

[CLOONAN] – “Competitive Analysis of Adaptive Video Streaming Implementations”; T. Cloonan Ph.D., Jim Allen Ph.D., The Cable Show NCTA/SCTE Technical Sessions spring 2011

[ULM] – “Adaptive Streaming – New Approaches for Cable IP Video Delivery”; J. Ulm, T. du Breuil, G. Hughes, S. McCarthy, The Cable Show NCTA/SCTE Technical Sessions spring 2010

[WHITE] – “Reclaiming Control of the Network from ABR Video Clients”; G. White, J. Ulm, The Cable Show NCTA/SCTE Technical Sessions 2012

APPENDIX – TEST SETUP DETAILS

Table 1 shows the details of the iOS client configuration and the media streams retrieved by the clients. Two iPads were connected to CM-4; one retrieving the complex 720p30 Football video and the other retrieving the less complex 480p30 News clip loop. Single iPads were connected to CM-3 and CM-1 with CM-3 delivering the complex 480p30 Dance Video clip and CM-1 streaming the complex 720p30 Football sequence. CM-5 had three iPads associated to it; each receiving 480p30 sequences with one of high video complexity and two of lower complexity. One iPod was associated to 2.4 GHz CM-2 and retrieved a low complexity 480p30 Soap Opera sequence while two iPods were associated to 2.4 GHz CM-6 with one iPod getting a less complex 480p30 black-and-white (B&W) TV clip and the other pulling the complex 480p30 Dance Video sequence.

Test configuration 1 had the goal of determining the baseline performance of unmanaged HLS clients retrieving the various media content over a DS BE SF as managed by the BSR2000 CMTS for each CM. The DS SF Maximum Sustained Traffic Rate (MSR) was set to 12 Mbps, the Maximum Traffic Burst (MTB) to 8 kBytes, and the Minimum Reserved Traffic Rate (MRR) to 0 Mbps.

Test configurations 2a and 2b entailed configuring per-HLS stream SFs so that clients retrieving the 720p30 content might be differentiated from those retrieving 480p30 content. To that end, test configuration 2a entailed applying SF MSR of 2.5 Mbps to the 480p30 content, which had maximum bit rate variant of 2.2 Mbps, and MSR of 4.5 Mbps to the 720p30 content which had maximum bit rate of 4 Mbps. Test configuration 2b doubled these MSR values to 5.0 Mbps for 480p30 and 9.0 Mbps for 720p30 clients. As before, MRR was set to 0 Mbps and MTB to 8 KBytes.

The tests were conducted by launching the iOS client Safari browser and pointing it to the URL of the desired media's variant playlist on the unmanaged HLS server connected behind the CMTS.

The PSNR is calculated as follows. Decoded video was subtracted from the uncompressed video. This difference provided the compression noise (distortion) present in each pixel of the video at that particular rate. Mean Square Error (MSE) was calculated by squaring the difference and averaging over the picture. As the maximum luminance value is 255, PSNR was calculated by taking the ratio of square of 255 and MSE. This was converted into dB by taking log of PSNR and multiplying it by 10. As this measures the compression noise, it provides a rough measure of visual quality. Higher is the PSNR, lower is the compression noise and roughly better visual quality.

Client No.	iOS Device Type	iOS Model	iOS Version	Cable Modem	SBG6580 WiFi Channel	ABR Media Stream	Video Complexity & Stream Rates (Mbps)
1	iPad 4	MD513LL	6.1.2	CM-4	157	Football 720p30	High @ 1,2,3,4
2	iPad 2	MC769LL	6.0.1	CM-4	157	News 480p30	Low @ 0.65, 1.2,1.7,2.2
3	iPad 2	MC705LL	6.0.1	CM-1	48	Football 720p30	High @ 1,2,3,4
4	iPad 1	MB292LL	5.0.1	CM-3	153	Dance Video 480p30	High @ 1,2,3,4
5	iPad 1	MB292LL	5.1.1	CM-5	36	Documentary 480p30	High @ 1,2,3,4
6	iPad 2	MC769LL	6.1.3	CM-5	36	Soap Opera 480p30	Low @ 0.65, 1.2,1.7,2.2
7	iPad 2	MC769LL	6.0.1	CM-5	36	Animation 480p30	Low @ 0.65, 1.2,1.7,2.2
8	iPod	MC540LL	5.1.1	CM-2	6	Soap Opera 480p30	Low @ 0.65, 1.2,1.7,2.2
9	iPod	MC540LL	5.0.1	CM-6	1	B&W TV Show 480p30	Low @ 0.65, 1.2,1.7,2.2
10	iPod	MC540LL	5.0.1	CM-6	1	Dance Video 480p30	High @ 1,2,3,4

Table 1. Adaptive Bit Rate HLS Client & Video Stream Details

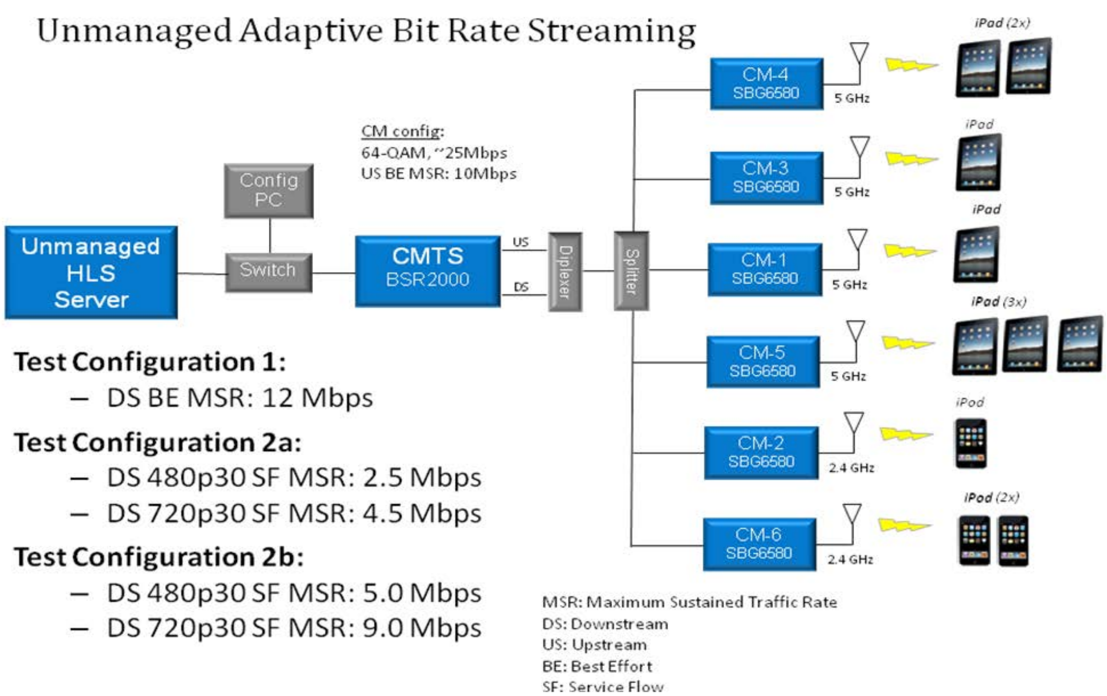


Figure 5 Adaptive Bit Rate Streaming Test Setup