IMPACT OF DELIVERY ECO-SYSTEM VARIABILITY AND DIVERSITY ON INTERNET VIDEO QUALITY

Aditya Ganjam, Prashanth Pappu, Ion Stoica, Jibin Zhan, Hui Zhang

Conviva, USA

ABSTRACT

We study the impact of the variability and diversity of multiple Internet delivery eco-system components (CDNs, ISPs, devices, video players, streaming protocols) on the quality of Internet video. Our study is based on a data set that is unique in two aspects: (a) *client-side instrumentation*: this provides critical insights into video quality as the video has traversed all eco-system components before reaching the client; (b) *large-scale*: our data set spans scores of popular sites, billions of streams, hundreds of thousands of video objects, and hundreds of millions of unique viewers. We show that there is large variability and diversity in each of the many Internet video delivery eco-system components, which can significantly impact the quality of video that is delivered to the end users. We present a set of architectural principles that are essential to optimize video quality in the presence of variability and diversity.

INTRODUCTION

Internet, as a distribution mechanism for video, presents huge opportunities for publishers and consumers. In particular, Internet's *global reach* enables publishers to connect with consumers any where without the geographical constraint of the traditional physical access networks such as cable networks, broadcast networks, and satellite networks. Content publishers such as Netflix, HBO, BBC, and CNTV take advantage of this by launching Internet video services in multiple markets around the globe. In addition, Internet allows consumers to access content via multiple devices including PCs, smart phones, pad devices, and TVs, i.e. 1st screen, 2nd screen, and 3rd screen. Finally, Internet video enables publishers to introduce rich social, interactive, and on-demand features.

To realize these benefits, a number of sophisticated technologies are used to implement a complex Internet delivery system. **Error! Reference source not found.**Figure 1 shows the typical end-to-end delivery pipeline for video delivery in the Internet. In a nutshell, once the video is captured, it is encoded and published to one or more video origin servers. From here, the video is either pushed or pulled by Content Distribution Networks (CDNs) and delivered over the Internet to client devices, where it is decoded and played back.

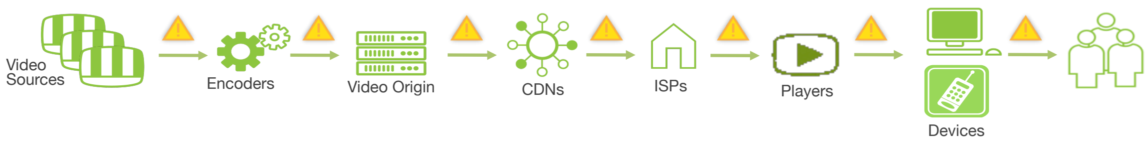

Figure 1: Encoding to payback end-to-end delivery chain for video

If any of the modules in this pipeline experiences issues, the quality will be impacted. The probability of failure is exacerbated by the huge diversity of devices, protocols, and systems that take part in the delivery, and the fact that different components are run and managed by different administrative entities. For example, typically, a content publisher uses different CDNs and ISPs to deliver video to different devices and players that are all built and managed by different vendors.

In this paper, we study the impact of the variability and diversity of multiple Internet delivery eco-system components on the quality of Internet video. In particular, we examine the following key components: CDNs (e.g., Akamai, Limelight, L3), ISPs (e.g. ATT, Comcast, BT, Telefonica), consumer devices (e.g. PC, iPhone, iPad, Xbox), software video players (e.g. Brightcove, OSMF, Platform), and streaming protocols (e.g. RTMP, Akamai HD1, HDS, HLS, SmoothStreaming). Our study is based on a large data set that has been gathered with a highly scalable data collection system over billions of streams, hundreds of thousands of video objects, and hundreds of millions of unique viewers. The system consists of two parts: (a) a client-resident instrumentation library in the video player, and (b) a data aggregation and processing service that runs in data centers. The client library gets loaded when Internet users watch video on an instrumented site. The library listens to events from the video player and additionally polls for statistics from the player. Because the instrumentation is on the client side we are able to collect very high fidelity raw data that captures the video quality after video has traversed all the eco-system components. We collect and process 1TB of data on average per day from various content providers over a diverse spectrum of end users, video content, Internet service providers, content delivery networks, and geographical regions.

Performance Metrics and Methodology

For the analysis in this paper, we first compute the performance for each video session. While there are several metrics that can be used to characterize the performance of a video session, we focus on (a) buffering ratio, which is defined to be the total amount of time a video player is spent in a re-buffering state (play-out buffer in empty and no video is showing on the screen) divided by the total video session duration; (b) the average of the bit rate of the session. Most of the video sites today implement adaptive streaming technology in which the video is encoded in multiple bit rates and a video player can dynamically pick a bit rate.

We pick these two per session metrics for the following reasons. First, as shown in an earlier study [3], buffering ratio is the most important metric with respect to impact on user engagement. Second, as users demand more HD quality content, the goal of adaptive streaming technology is to have a user to view the content at the highest bit rate video subject to the capacity availability of ISP, CDN, and device at the moment. In general, it is difficult to achieve both lower buffering ratio and high average bit rate simultaneously.

We compare two instances of one eco-system components by computing the aggregate metrics for sessions that pass through the components. For example, when we compare the performance of two CDNs, we use the average buffering ratio across all the sessions that are streamed from each of the two CDNs.

Variability of Content Delivery Network (CDN) and ISP Performance

Today video content publishers use a variety of CDNs, such as global CDNs, regional CDNs, operator CDNs, and in-house CDNs. Each of these CDN types varies in its reach, peak performance, support for delivery protocols and other characteristics. While most CDNs offer standard streaming protocols, some emphasize proprietary protocols further increasing diversity.

- **Global CDNs:** These CDNs are the largest, provide the widest coverage, and support most of the standard protocols and content protection technologies. However, their deliver quality can vary on per country basis, as the amount of investment in the infrastructure and ISP peering varies.
- **Regional CDNs:** These CDNs focus on providing service in certain geographic regions (e.g., Northern Europe). They are not as large as global CDNs, and may focus on specific market segments (e.g., live events).
- **Operator CDNs:** These CDNs are run by ISPs. While they are typically cheaper than other CDNs, their delivery is restricted to the ISP's users. Operator CDNs are fairly new to the market, and have not built the level of reliability and support capabilities that Global and Regional CDNs have. The streaming protocols and content protection schemes they support may vary across operators.
- **In-house CDNs:** These can range from a set of servers to a fully distributed CDN infrastructure and are operated by the content publisher. They are feature limited, as they specifically target the needs of the content publisher.

ISPs are indispensable links in the video distribution chain. Unfortunately, the performance of the ISPs can vary dramatically, due to an array of factors, such as different communication technologies, service models, and peering strategies. For example, wireless ISPs tend to offer less stable and lower capacity connections than their wired counterparts. Even two wired ISPs can have different performance profiles as they use different datalink technologies (e.g., DSL vs. cable modems). Using the same technology does not guarantee the same streaming quality either, as some ISPs may use traffic limiting, which can significantly impact video quality. Finally, direct peering with top CDNs enables ISPs to provide better quality by avoiding transit through other networks.

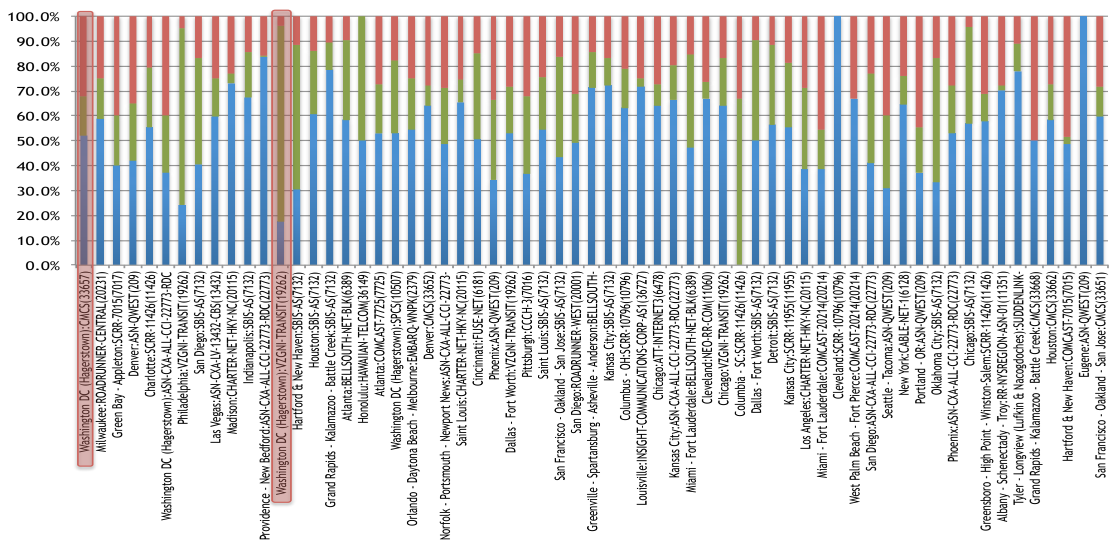

Figure 2: Performance Variability of CDN Performance across ISPs

Figure 2 illustrates the variability in performance across both CDNs and ISPs. Each column corresponds to an ISP, and each stacked bar within a column corresponds to a CDN. The height of each bar within a column shows the percentage of time for which that CDN has outperformed the other two. There are two points worth noting. First, the performance of every CDN can vary widely from ISP to ISP. Second, there is no single CDN that outperforms the other two for all ISPs. In particular, while in many cases the blue CDN outperforms everyone else, there are quite a few cases in which the green CDN provides the best performance.

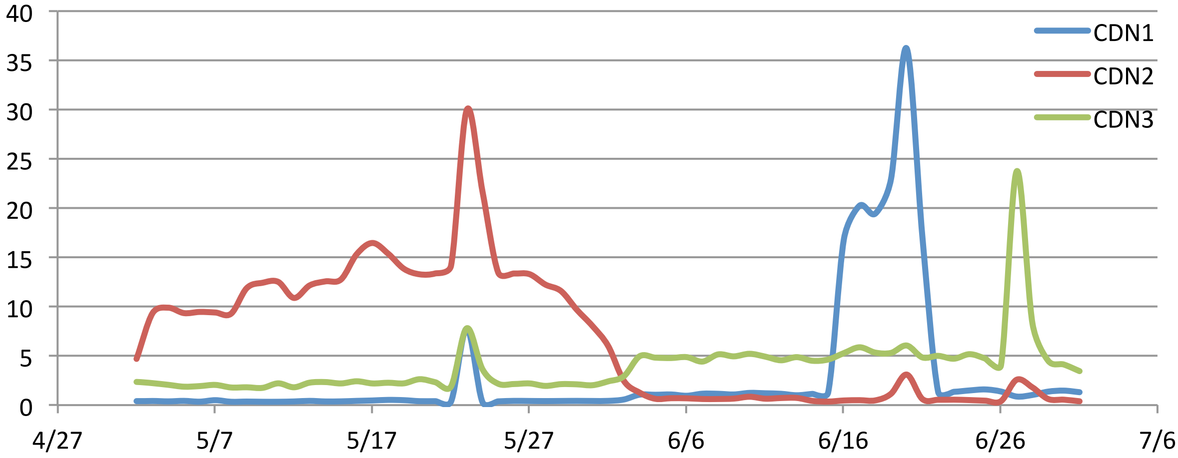

Figure 3: Performance Variability of CDN Performance (Streaming Error %) Across Three Months Period

Next, we illustrate the variability of CDN performance in time. Figure 4 plots the percentage of streaming errors over a two months period three CDNs. Again, there is no single best CDN, and at different moments of time there are different CDNs performing the best or the worst. This suggests there is a significant potential of improving the streaming quality by dynamically picking the best CDN at any given time. While in general the performance of the CDNs is not correlated, there is one moment of time (around 5/21-5/22) when all CDNs experience spikes in the streaming errors. This corresponds to a high profile event that overloaded the origin causing failures on all CDN.

Diverse Devices and Video Players

When it comes to devices and video players we are witnessing an even higher heterogeneity than in the case of CDNs and ISPs. A typical content publisher supports a wide array of devices and players to enable video viewing ‘anytime and anywhere’ on PCs, phones, and TVs. Devices can vary significantly in capabilities (e.g., support for adaptive bit rate, support for content protection, etc.), performance (e.g., memory footprint of apps, CPU usage), and in the development environment (e.g., programming language). Table 4 shows the capability along three key dimensions across several popular devices.

Device / Platform	Supported Streaming Protocols	Supported Content Protection	Development Language
PC / Flash	RTMP, HDS, HTTP progressive download	URL Tokenization, Flash Access DRM, SWF Verification, Protocol encryption	ActionScript

iPhone / iOS	HLS, HTTP progressive download	Encryption, Apple DRM and some 3rd party DRMs	Objective-C
Xbox	SmoothStreaming HTTP progressive download	PlayReady DRM	C#
PC / Silverlight	SmoothStreaming, MMS, HTTP progressive download	PlayReady DRM, URL Tokenization	C#
HTML5 Video	HLS, HTTP progressive download	No standard content protection	Javascript
Android	HTTP progressive download, HLS (3.x and later)	Encryption, some 3rd party DRMs	Java
Roku	HLS, HTTP progressive download	Encryption	Brightscript

Table 4: Heterogeneity in device capability

Differences in streaming protocol and content protection usually mean the content publisher must prepare multiple copies of content. Differences in development language and player development frameworks mean the content publisher must build different applications. In addition, lack of standardized support for streaming protocols or content protection on a device leads to proliferation of third party solutions or lack of adoption. Android is a good example of the first case. Android only supports HLS in the standard player starting from version 3, which is only used on tablets. A majority of Android phones in the market at version 2.x and require a third party solution to play HLS content. This quickly resulted in several third party solutions for Android further complicating the delivery chain. A good example of the second scenario where the platform is not adopted is HTML5 video. While HTML5 video is gaining significant adoption in general, it lacks adoption by premium content publishers due to the lack of content protection.

Performance of different platforms also varies significantly, depending on the maturity of the platform, protocols and technologies available on the platforms, etc. Figure 5 shows that for the same content provider, buffering ratio on different platforms is quite different from each other. Buffering ratio represents the percent of time viewers spend in a re-buffering state.

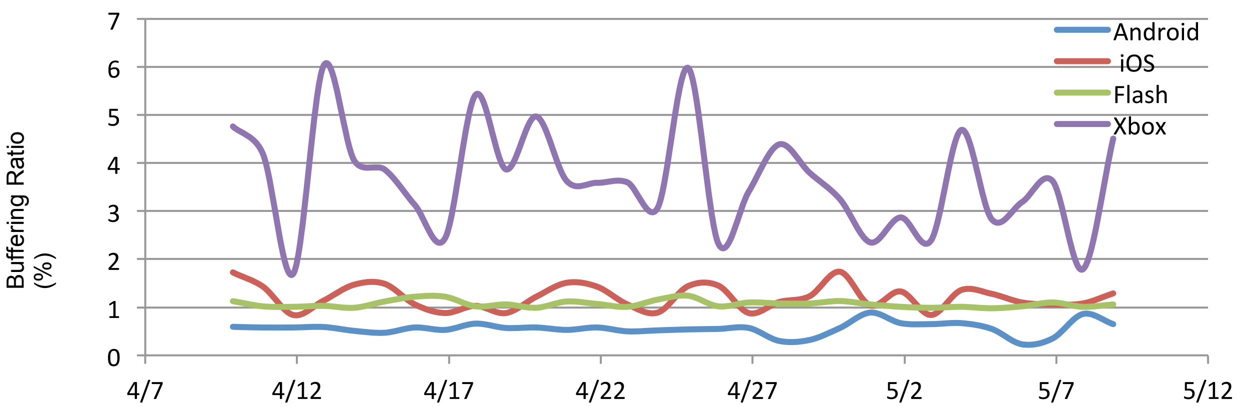

Figure 5: Performance Variability (Buffering Ratio) Among iOS vs. Flash vs. Xbox vs. Android Players

The quality of the picture shown to viewers also varies between players on the sample platform. . Figure 6 shows the percentage of views with over 1 Mbps average bit rate on the five different Flash based video players. All five video players are production players for major premium video web site. All five implement adaptive streaming with multiple video bit rates. However, they differ in their adaptive streaming *algorithm:* when to switch up or down a bit rate and which bit rate to switch to. As discussed earlier, it is fundamentally difficult to achieve both high average bit rates and lower buffering ratio. One way to avoid buffering is to be very conservative in switching up to high bit rates, thus resulting in lower average bit rate.

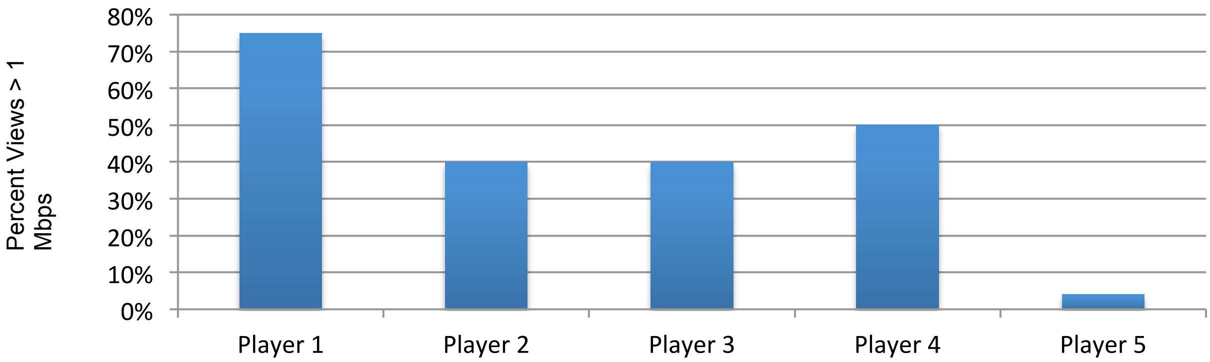

Figure 6: Performance Variability Between Video players in Flash (percentage of HD streams)

Streaming Protocols

Content publishers use a variety of streaming protocols to accommodate the diversity of devices and video formats. Streaming protocols fall into two categories: (1) stateful client-server protocols and (2) stateless HTTP-based protocols. Examples of stateful protocols are RTMP (developed by Adobe for Flash), MMS (developed by Microsoft for Windows Media Player), and RTSP. Since these protocols require server-side support, CDNs need to deploy custom servers to support these protocols (e.g., Adobe's FMS or Microsoft's Media Servers). In contrast, stateless HTTP-based protocols chunk the video in small pieces that are distributed as regular files through the existing HTTP infrastructure. Examples of such protocols are Apple's HLS, Adobe's HDS, and Microsoft's Smooth Streaming. While these protocols can leverage the existing HTTP servers for distribution, they typically incur a higher end-to-end latency.

The diversity of these protocols increases the complexity of the video distribution ecosystem, especially as different devices use different streaming protocols. For example, today, the common streaming protocol on PCs is RTMP, on iOS devices is HLS, and on Microsoft's devices is Smooth Streaming. While there are standardisation efforts to migrate to MPEG-DASH, we expect this fragmentation to continue for the next few years.

Protocol	Buffering Ratio	View > 1 Mbps
RTMP	1.0%	59.4%

AKHD1	4.6%	56.9%
AKHD2	1.5%	46.0%
HLS	1.6%	17.5%
Smooth Streaming	2.3%	77.2%

Table 7: Performance Variability between streaming protocols

Table 7 shows some performance metrics of different streaming protocols

ARCHITECHCAL PRINCIPLES FOR OPTIMIZING VIDEO QUALITY IN THE PRESENCE OF VARIABILITY AND DIVERSITY

As has been discussed in the previous sections, the many components in the Internet video distribution eco-system exhibit large variability and diversity that can result in significant degradation of video quality. As Internet video is becoming better monetized and is increasingly consumed on large screens, consumers' expectations for high quality will continue to increase. At the mean time, while we expect continuous improvement of technology in all components of the Internet distribution eco-systems, variability and the diversity will take different forms but will remain to be key challenges for predictable, consistent, and high quality video experience as demanded by both content publishers and consumers.

To insulate content publishers and consumers from this continuously evolving variability and diversity, a powerful service-layer software is needed to optimize the video quality in the presence of sophisticated delivery eco-systems with diverse and variable components. The detailed design of such a service-layer software is beyond the scope of this paper. However, we do believe that any such service-layer software should incorporate the following architectural principles:

1. The system should take advantage of the diversity and variability of the delivery eco-system to solve the quality problem. For example since there are usually independent failure modes among different CDNs, servers, and ISPs, content should be distributed via multiple CDNs. Each client video player should retrieve video from multiple servers in different CDNs via diverse Internet paths. This concept of taking advantage of diversity to achieving predicable and consistent quality is a powerful technique used to solve other computer science problems. For example, the modern high available Internet services are built on top of large number of servers, each of which does has performance variability and failure modes. The traditional architecture of having a single CDN to distribute video is inadequate.
2. The optimization mechanism should be placed in the client-side, NOT the server-side, to enable the client to dynamically select bit rate, CDN, server, or Internet path. The traditional architecture of using server-side mechanism is inadequate.

3. The optimization should be continuous throughout the duration of a video session. The traditional architecture of DNS re-direction and DNS load balancing at the beginning of a session is inadequate.
4. The optimization should be dynamic and based on the real-time condition of various eco-system components such as ISPs and CDNs. The traditional architecture of using DNS re-direction and DNS load balancing based on stale network state information is inadequate.
5. The optimization should be specific for each individual video player. The traditional architectural of optimization based on average network state information is inadequate.

SUMMARY

The desire to achieve 'anytime, anywhere' video distribution imposes great challenges on Internet video publishers. A publisher has to deal with multiple encoder formats and profiles, CDNs, ISPs, devices, and a plethora of streaming protocols and video players to ensure accessibility of video to all viewers. We have shown that there is significant variability in each of these diverse components and together these variability and diversity present huge challenges to a consistent, predictable, and high quality user experience. We argue that this quality challenge can be addressed by taking advantage of the diversity. In particular, the architectural principles are the following: (1) multiple CDNs should be used for video distribution, (2) the optimization mechanism should be placed at the client instead of server, and (3) the optimization should be continuous throughout the duration of a video session, based on the real-time performance of eco-system components, and specific to each individual video player.

REFERENCES

1. Global Internet Phenomena Report: Spring 2011, http://www.sandvine.com/news/global_broadband_trends.as

2. Cisco Visual Networking Index: Forecast and Methodology, 2009-2014, http://tinyurl.com/3p7v28,

3. F. Dobrian, A. Awan, I. Stoica, V. Sekar, A. Ganjam, D. Joseph, J Zhan, and H. Zhang, 2011. Understanding the Impact of Video Quality on User Engagement. Proceedings of SIGCOMM 2011.

4. S. Ali, A. Mathur, H. Zhang. Measurement of Commercial Peer-to-Peer Live Video Streaming Workshop on Recent Advances in Peer-to-Peer Streaming 2006.

5. X. Hei, C. Liang, J. Liang, Y. Liu, K. W. Ross. A Measurement Study of a Large-scale P2P IPTV System. In Proceedings IEEE Transactions on Multimedia 2007.

6. A. Mahimkar, Z. Ge, A. Shaikh, J. Wang, J. Yates, Y. Zhang, Q. Zhao. Towards Automated Performance Diagnosis in a Large IPTV Network. In SIGCOMM 2009.

7. K. Sripanidkulchai, B. Maggs, H. Zhang. An Analysis of Live Streaming Workloads on the Internet. In Proceedings of IMC 2004.

8. H. Y. et al. Inside the Bird's Nest: Measurements of Large-Scale Live VoD from the 2008 Olympics. In Proceedings of IMC 2009.