Revisiting a QoE Assessment Architecture Six Years Later: Lessons Learned and Remaining Challenges

Amy Csizmar Dalal*

Carleton College, One North College St., Northfield MN 55057, USA
adalal@carleton.edu

Abstract. In 2003, we presented an architecture for a streaming video quality assessment system [1]. Six years later, many of the challenges outlined in that paper remain. This paper revisits the 2003 architecture, updates it given what we have learned in our experience thus far with developing the architecture, and discusses in detail the remaining challenges to the realization of this architecture. We conclude with suggestions for moving beyond the biggest challenges, namely cooperation among the interested parties and system scale.

Key words: Quality of Experience (QoE), Quality of Service (QoS), Streaming Media, Measurement, Performance, Reliability

1 Introduction

In 2003 we published a paper [1] proposing a new architecture for assessing users' quality of experience (QoE) of streaming video, video sent on-demand over unicast from a media server to one or more media clients. At the time, RealPlayer was the dominant mechanism for video delivery over the Internet, and the amount of video traffic was a small but increasing fraction of overall Internet traffic [2]. Today, video makes up about 30% of Internet traffic [3], with much of the video embedded on sites such as YouTube and Hulu [4]. Then and now, content providers, content distributors, and ISPs are interested in determining how current network conditions, such as packet losses and delays and available bandwidth, affect the performance, as perceived by the end-users, of streaming video applications, and vice versa. It is this user perception, after all, that drives Internet, application, and content use.

The notion of QoE is a nebulous one: there is no one accepted definition for QoE of streaming video, nor an accepted standard of streaming video QoE measurement. While subjective approaches, such as the Mean Opinion Score [5], are the most natural choice, they suffer from scalability and context accuracy issues. Indeed, a survey of the literature shows that QoE has been measured using

* This work is sponsored by grants from the Howard Hughes Medical Foundation and from Carleton College. Early versions of this work were sponsored by Hewlett-Packard Laboratories.

network-level measurements [6, 7, 8], frame rate [9], packet-level statistics [10], received signal strength indicators [11], and a complex combination of network, application, and content measurements [12].

The breadth of QoE measurement approaches speaks to the complexity of the QoE measurement problem. Yet we must overcome these challenges to create robust, dependable, efficient, and effective QoE measurement systems. Such systems will move the current trial-and-error approach of network provisioning, application deployment, and protocol design and development to a place where application performance combined with knowledge of network conditions leads to a more responsive approach to video delivery over the Internet, and to future Internet support of rich media applications, such as high-definition video for entertainment, distance learning, telemedicine, and telepresence.

In this paper, we revisit our proposed architecture for a QoE assessment system for streaming video. The main focus of our work to date has been on developing the measurement tool and appropriate QoE measurement(s). We discuss the challenges in developing a QoE measurement, revisit the original architectural goals, update the architectural design, and discuss the remaining challenges. Many of the system-wide challenges that we wrote about in 2003 still exist, and some of them, like garnering cooperation among the interested parties, provide significant barriers to realizing this architecture.

We start in Section 2 by discussing the original architecture goals and describing the motivations and barriers to cooperation of the involved entities. Section 3 highlights the process and remaining challenges in selecting a QoE measurement. Section 4 updates the architecture and discusses the new design elements, including the design of a new key aspect of our system, the health monitor. Section 5 concludes by discussing the key remaining challenges and proposing mechanisms for moving towards the realization of this architecture.

2 Original Architecture

Figure 1 illustrates our original QoE assessment architecture, which focuses on four main goals:

1. **Flexibility** Measurements should be collected automatically and with a minimal amount of end-user participation. Additionally, the architecture should support on-demand, non-disruptive measurements from any subset of participating hosts, either for diagnostic or planning purposes. The assessment servers, which control the collection of data from media clients and helper agents and coordinate the analysis of this data, provide this functionality within the architecture.
2. **Support for multiple consumers of assessment** Multiple parties have a vested interest in supporting and delivering streaming video. A stream quality measurement system should provide a mechanism for these parties to fully participate in the system by both sharing and receiving data from other

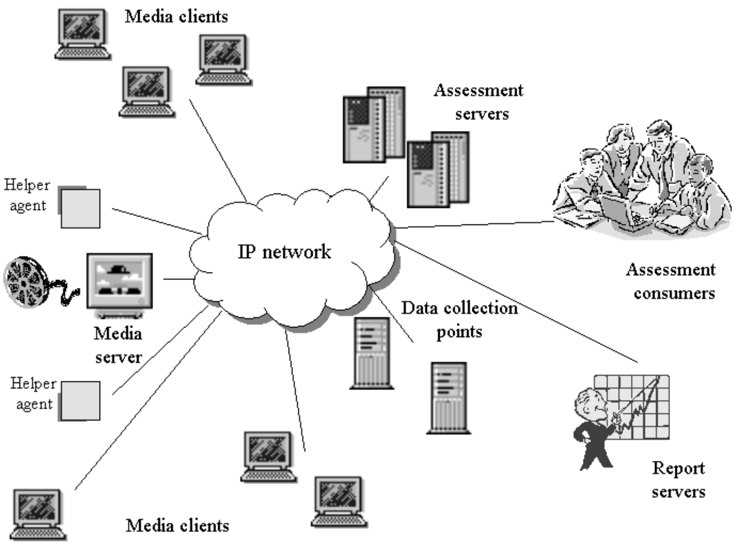

Fig. 1. The original QoE assessment architecture, from [1].

parties. The report servers, which disseminate analyzed data, accomplish this architectural goal.

3. **Utilization of existing infrastructure** A large-scale QoE assessment architecture should leverage existing applications and infrastructure wherever possible, including other measurement infrastructures and existing tools, to reduce the burden on end-users and the network. The proposed architecture accomplishes this by the use of helper agents, which may be existing measurement points within the network, and by collecting measurements from the media player applications (see Section 3).
4. **Responsiveness** QoE measurements should be used to modify how the system behaves. Examples include determining whether network or server resources should be reprovisioned, whether a stream should be served from a different location on the network, or whether a server should temporarily decrease or increase its sending rate. In the proposed architecture, the report servers and assessment servers collaborate to make this happen.

A QoE assessment system for streaming video necessarily involves independently-operating independently with common needs and goals. **Media servers**, for instance, are most interested in knowing if they are streaming at a bandwidth that their clients can support. **Media clients**, on the other hand, want assurance that their stream quality will either not degrade, if it is currently acceptable, or improve soon, if it's currently less than acceptable. They may also want to know if another media server can deliver the same content at a better quality level. **Content distributors** are interested in content placement for optimal performance, as well as how many servers are needed and whether transcoding of content is necessary. **Network operators** want to know if their networks are healthy, which paths are over- or under-provisioned, whether better routes exist to certain destinations, and whether reprovisioning of resources is necessary. **Network architects** are interested in how to provision and design networks to best support video traffic, including router placement and peering relationships. Finally, **protocol developers** and **application designers** are interested in how to best utilize existing infrastructure to maximize performance, and in knowing what network and/or application conditions are most likely to affect performance.

There are points along the delivery system where cooperation is required. A content distributor cooperates with network operators and ISPs in order to determine where to locate content servers. Media servers and media clients interact with ISPs to connect to the Internet. Other cooperative relationships, if they existed, would prove beneficial to both parties. Application designers could work with network operators to improve how applications utilize existing network infrastructure. Similarly, network architects could utilize feedback from media servers and clients to improve the resource distribution of future networks.

Historically, cooperation among these parties has been limited. Given a set of data from, for example, a network provider, it is often trivial to determine the source, destination, and/or nature of the data traffic. This opens up both privacy and trade secret issues: customers may not want their competitors to know what sites they are visiting; an outsider may be able to infer the topology of a network, or an ISP's peering agreements; knowledge of an ISP's customers may violate privacy agreements and give its business rivals a competitive advantage. Measurement data might also be used to design a more effective denial of service attack against an ISP or set of media servers. Finally, there is the fear that measurement data might demonstrate that an ISP's service level agreement is not being met for one or more of its customers, opening it to lawsuits for breech of contract. These represent real barriers to cooperation, and we recognize that these issues are non-trivial and difficult to fairly address. We return to this point in Section 5 and present several ideas for mitigating these barriers.

3 QoE Measurement

A key challenge in the development of a QoE measurement architecture is defining QoE for streaming video. A survey of the literature yields several definitions: video distortion as seen by the end user [13]; the number and severity of impairments caused by transmission parameters such as noise [14]; how a user would rate the subjective quality of the stream as compared to other streams s/he has viewed in the past [15, 6]; how close the video quality is to satellite or cable video quality [8]. To date, there is no measurement standard for QoE for streaming video, as we discuss in Section 1.

Our approach is to exploit the ease of collection and analysis of objective measurements and infer the user's experience by collecting measurements as close to the user as possible, at the application layer, using an instrumented version of a media player application [15]. The application layer measurements are composed of stream state information reported by the player, which we poll once per second, on quantities such as current average bandwidth, number of lost and retransmitted packets, frame rate, and number of times buffered. They reflect the current state of the video stream as well as the media server's response to network conditions, such as packet losses. From this state information, we can infer what a user "sees", particularly if we know the expected state of a "normal" stream.

As we discuss in [10], not all state measurements are created equal: some of them, such as retransmitted packets, are early indicators of the presence of network congestion; while others, such as average bandwidth, are lagging indicators of network congestion. [2] Because a goal of our architecture is to *predict* streaming video QoE, we focus on collecting and analyzing leading indicators, specifically the number of successfully retransmitted packets.

Discerning the QoE of a video stream requires knowledge of the QoE of past streams. Users are likely to assign similar QoE ratings to streams with similar stream state characteristics.Thus, a measurement system could compare past stream state data and QoE ratings against stream state measurements of a currently-streaming video to assign a QoE rating to that video. Data mining techniques can be used to efficiently search this historical data to find the closest matching stream, and assign a QoE rating to the current stream based on the rating of the closest match. This is the idea behind our own QoE measurement strategy. As a proof of concept, we collected MOS-like measurements and stream state measurements for 228 streams and applied a nearest-neighbor data mining technique to this data to predict stream QoE ratings. Our experiments showed that this approach results in correct QoE ratings assignments between 70 and 90% of the time; see [15, 16] for more details.

Collecting measurement data on a large enough scale to demonstrate the accuracy of various proposed QoE measurements remains a challenging task; most existing data results from smaller experiments on testbed networks, with possibly unrealistic network conditions. There are also questions as to whether a generic QoE measurement can exist for different video scenarios, such as live versus on-demand content or streaming versus progressive download. If different QoE measurements are required by different applications, the architecture must be nimble enough to switch between these measurements and analyses on the fly. Finally, there is the question as to whether measurement and analysis can be completed on sufficient time scales to allow affected parties to react and respond to the results. Our results in [16] show promise, but further work must be done for this architecture to prove viable.

[2] To some extent, whether a measurement is a leading or lagging indicator of network congestion depends on the state information that the media player reports. Windows Media Player, for instance, reports the number of application packets that were successfully retransmitted, while QuickTime does not.

4 System Architecture

Figure 2 updates the architecture discussed in Section 1 based on our experience in developing parts of the system architecture. The new architecture better reflects the complex relationships along the critical path of measurement, and better meets the goals of responsiveness and support for multiple consumers of assessment. This version of the architecture also focuses more specifically on the QoE measurement mechanism.

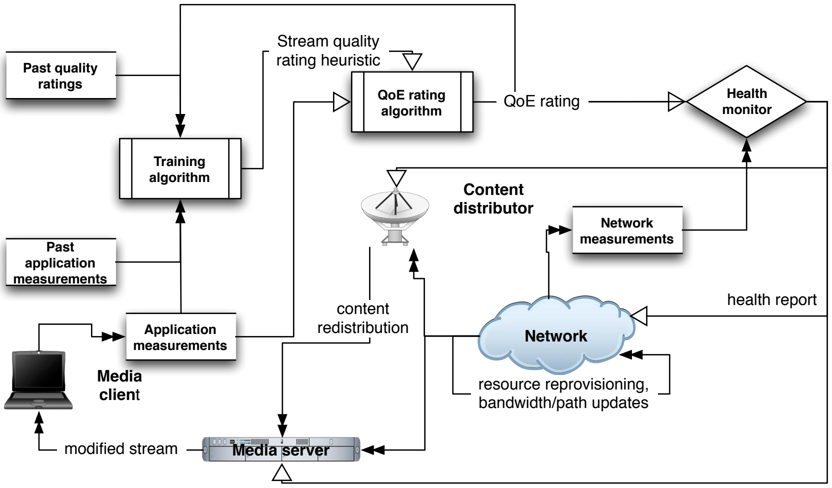

Fig. 2. The revised streaming video QoE assessment architecture. Forward paths are denoted by lines with white arrows, while feedback paths are denoted by lines with double black arrows.

The revised architecture introduces the QoE rater and the health monitor, which utilize current and past measurements to infer current QoE and system health levels, respectively. The QoE rating algorithm relies on past and current application measurements, as well as past QoE ratings, to assign QoE ratings to current streams. The training algorithm uses past QoE and stream state measurements to develop a heuristic by which to evaluate the QoE of current streams, which is used by the QoE rater. The health monitor analyzes the QoE ratings, as well as current network measurements, and periodically sends out status reports to the system entities. The health monitor can also send more frequent updates in cases where it detects that the health of the system is deteriorating from an acceptable level.

Each entity acts upon the information received from the health monitor however it sees fit. Each entity can also request information from other parties in the system. For instance, content distributors may match up health monitor reports with available bandwidth reports from network operators to determine when and how to redistribute content. Media servers might also utilize bandwidth and path updates from network operators to optimize their streaming rates. As updated application and network measurements are fed back into the system, the system becomes self-supporting, reflecting both the current state of the network and the relevant history of the network and application states, similar to the system proposed in [17].

The RTP feedback architecture [18], RTCP, shares many of the same goals we seek here, and yet has historically suffered from scalability and usability issues. Unlike RTCP's feedback mechanism, which originates solely from the client, feedback in our architecture originates from multiple sources, allowing for a more holistic consideration of system state. Also, while RTCP sends all reports upstream, our architecture allows for multiple feedback and feed-forward paths, to allow for continuous monitoring and updating of system state.

Because the health monitor and QoE rater constantly respond to new measurement data, the system necessarily evolves as application and network data evolve, and as traffic patterns and application usage patterns change. Thus, the system can support both current and future streaming video applications and traffic patterns, and can evolve along with the Internet and its applications.

While the structure of the health monitor is quite simple, its implementation poses several key challenges. Defining an appropriate time-scale for measurement reporting requires a tradeoff between accuracy and relevancy: the monitor needs to distinguish between normal and troublesome network behavior and recognize diurnal and weekly patterns in the data, but must also be nimble enough to respond to sudden changes in network state, such as router outages or server overload, and sudden changes in QoE levels, such as when a media server goes down. We propose a two-pronged strategy. The health monitor receives periodic, time-averaged measurements from the network under "normal" circumstances: for example, packet loss rates over the last five minutes. This reduces unnecessary communication between the network measurement entity and the health monitor. Instantaneous, more frequent measurements are allowed under certain conditions, determined either by the network measurement entity, the health monitor (in the case where QoE ratings drop sharply, for instance), or both. Identifying adequate time periods for more frequent measurement could be determined on a case-by-case basis or by trial and error among the parties. This strategy prevents the system from being overly sensitive and from being unresponsive. An additional challenge is involved in determining when to examine network measurement data and/or QoE ratings more closely. A simple solution is to only consider network measurements when the QoE rating itself is low, borderline, or decreasing from a previous steady state; or, conversely, to consider QoE levels only when network measurements imply declining network conditions. If the QoE measurement, or the network state, is acceptable and stable, no further action needs to be taken.

5 Remaining Challenges: Cooperation and Scale

In previous sections, we have discussed the evolution of our streaming video QoE assessment architecture in light of our experiences developing and realizing the architecture. So far, our work has successfully addressed two of the largest challenges: identifying an appropriate QoE metric and developing a means for analyzing QoE-related data. In order to fully realize the architecture, two key challenges remain: garnering cooperation among independent entities, and determining an appropriate system scale. We address these remaining challenges in this section.

Garnering cooperation entails addressing the main concerns about sharing data addressed in Section 1: privacy, security, trade secrets, and legal issues. Solutions already exist to deal with anonymizing and sanitizing network measurement data, although these may be imperfect [19]. Safeguards such as lifetimes could be imposed on shared data, after which the data could only exist in summary form, if at all. Standards of conduct, such as the best practices proposed in [20], should be more widely accepted and implemented. Legal agreements can be brokered between the key parties to address the legal and trade secret issues that can occur in the act of sharing data. For instance, ISPs, in exchange for sharing available bandwidth information with content providers, will receive data from the content providers and from the system to allow it to reprovision itself more efficiently. Content distributors, in exchange for sharing content and encoding information with various parties, would in turn receive path availability information from the networks that would allow them to better provision and place their servers.

Scale is another key challenge in the realization of this architecture. While ideally such an architecture would exist on a nationwide or international scale, in reality it is easier to garner cooperation and marshall the necessary resources on a smaller scale—for example, a content distribution network and its partners. Such a system could leverage the existing business relationships between content providers, the media servers that deliver the content, the connected ISPs and their network operations, and the content distribution networks, allowing for a more natural development of trust. Such closed systems, particularly early on in the deployment of this architecture, could very well provide the needed proof-of-concept for the viability and value of this architecture, paving the way for more wide-scale implementation, perhaps regionally or nationally.

As our experience over the past six years, and the work of the networking community for over a decade, has demonstrated, developing and implementing QoE assessment mechanisms for streaming video remains a difficult and sometimes vexing problem. We have made some key strides, particularly in identifying application-level measurements from which QoE can be inferred, but still need to find viable solutions for the cooperation and scale issues. While the challenges remain great, the payoff of a self-supporting, responsive, and self-healing system for streaming video delivery, in which the complex interactions between network and application performance are better understood, and in which this knowledge is leveraged to design and implement better networks, applications, and protocols, will be greater still.

References

1. Csizmar Dalal, A., Perry, E.: A new architecture for measuring and assessing streaming media quality. In: Proceedings of PAM 2003, La Jolla, CA (April 2003)
2. Li, M., Claypool, M., Kinicki, R., Nichols, J.: Characteristics of streaming media stored on the Web. ACM Transactions on Internet Technology (TOIT) **5**(4) (2005) 601–626
3. Business Wire: Ellacoya data shows web traffic overtakes peer-to-peer (p2p) as largest percentage of bandwidth on the network (June 18 2007) `http://www.businesswire.com/portal/site/google/index.jsp?ndmViewId=news_view&newsId=20070618005912&newsLang=en`.
4. comScore, Inc.: YouTube attracts 100 million U.S. online video viewers in October 2008 (December 9 2008) `http://www.comscore.com/press/release.asp?press=2616`.
5. P.910, I.T.R.: Subjective video quality assessment methods for multimedia applications. Recommendations of the ITU, Telecommunications Sector
6. Calyam, P., Sridharan, M., Mandrawa, W., Schopis, P.: Performance measurement and analysis of H.323 traffic. In: Proceedings of the 2004 Passive and Active Measurement Workshop, Antibes Juan-les-Pins, France (April 2004)
7. Cole, R.G., Rosenbluth, J.H.: Voice over ip performance monitoring. SIGCOMM Comput. Commun. Rev. **31**(2) (2001) 9–24
8. Tao, S., Apostolopoulos, J., Guerin, R.: Real-time monitoring of video quality in IP networks. IEEE/ACM Transactions on Networking **16**(5) (October 2008) 1052–1065
9. Wang, Y., Claypool, M.: RealTracer - tools for measuring the performance of RealVideo on the internet. Kluwer Multimedia Tools and Applications **27**(3) (December 2005)
10. Csizmar Dalal, A., Kawaler, E., Tucker, S.: Towards real-time stream quality prediction: Predicting video stream quality from partial stream information. In: Proceedings of The Sixth International ICST Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness (QShine), Las Palmas de Gran Canaria, Spain (November 2009) to appear.
11. Li, M., Li, F., Claypool, M., Kinicki, R.: Weather forecasting - predicting performance for streaming video over wireless LANs. In: Proceeedings of NOSSDAV, Stevenson, Washington (June 2005)
12. Gulliver, S.R., Ghinea, G.: Defining user perception of distributed multimedia quality. ACM Trans. Multimedia Comput. Commun. Appl. **2**(4) (2006) 241–257
13. Babich, F., D'Orlando, M., Vatta, F.: Video quality estimation in wireless ip networks: Algorithms and applications. ACM Transactions on Multimedia Computing **4**(1) (January 2008)
14. Boutremans, C., Iannaccone, G., Diot, C.: Impact of link failures on VoIP performance. In: NOSSDAV '02: Proceedings of the 12th international workshop on Network and operating systems support for digital audio and video, New York, NY, USA, ACM (2002) 63–71

15. Csizmar Dalal, A., Musicant, D.R., Olson, J., McMenamy, B., Benzaid, S., Kazez, B., Bolan, E.: Predicting user-perceived quality ratings from streaming media data. In: Proceedings of ICC 2007, Glasgow, Scotland (June 2007)
16. Csizmar Dalal, A.: User-perceived quality assessment of streaming media using reduced feature sets. Technical report, Carleton College (April 2009)
17. Allman, M., Paxson, V.: A reactive measurement framework. In: Proceedings of the Passive and Active Measurement Conference, Cleveland, OH (April 2008)
18. Schulzrinne, H., Casner, S., Frederick, R., Jacobson, V.: RTP: A transport protocol for real-time applications. IETF RFC 3550 (July 2003)
19. Pang, R., Allman, M., Paxson, V., Lee, J.: The devil and packet trace anonymization. Computer Communication Review **36**(1) (January 2006)
20. Allman, M., Paxson, V.: Issues and etiquette concerning use of shared measurement data. In: Proceedings of the ACM Internet Measurement Workshop, San Diego, CA (October 2007)