telchemy

Application Note

Title **Measuring the Quality of Video Streaming Services**

Series IP Video Performance Management

Date January 2021

Video streaming is an increasingly popular form of content delivery that offers benefits to both content providers and end users. This Application Note describes commonly used video streaming methods and protocols, and explains how objective measurement tools can be used to monitor and manage viewer QoE.

Contents

Introduction

Media streaming is a method of sending encoded multimedia content, usually audio and/or video, in packetized form across a network so that it can be viewed immediately by an end user. In contrast with progressive download, in which the entire media file is usually cached to a local directory as it downloads, streaming video content is temporarily buffered, played and then discarded by the media player. Video streaming can be used to deliver content on demand—for example, to play web clips or stream a movie/television show selected by an end user—or in real time, such as a live streamed webcast of a sporting event.

Video streaming offers several advantages over progressive download:

- Content providers have more control over the content; files are not stored locally, making them more difficult to copy.
- End users can skip to any point in a pre-recorded video without downloading the entire file first.
- Bandwidth may be used more efficiently; video can be encoded at a lower bitrate for delivery to users with slow connections.

Disadvantages of video streaming include:

- The need for a streaming protocol in addition to the network protocols normally required for file transfer.
- The requirement for a streaming server for some types of streaming, such as real-time delivery of live content using RTSP/RTP.
- The possibility that uses with extremely slow or unstable connections may be unable to view even low bitrate video reliably.

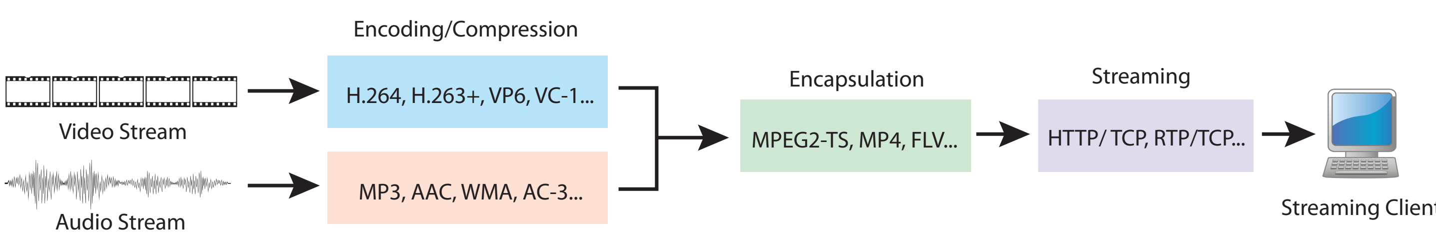
How Video Streaming Works

In a typical video streaming service, a video stream is encoded using a codec such as H.264, H.265 or VC-1, an audio stream is encoded using a codec such as MP3 or AAC, and both streams are encapsulated in a container format such as MPEG2-TS, MP4 or FLV. The container bitstream is sent from a server to a streaming client, typically using HTTP over TCP, HTTP over QUIC, or RTP over UDP or TCP. In most cases, the client communicates with the server using a streaming protocol such as RTSP, RTMP, DASH, or one of several proprietary streaming protocols that are described in the next section of this document.

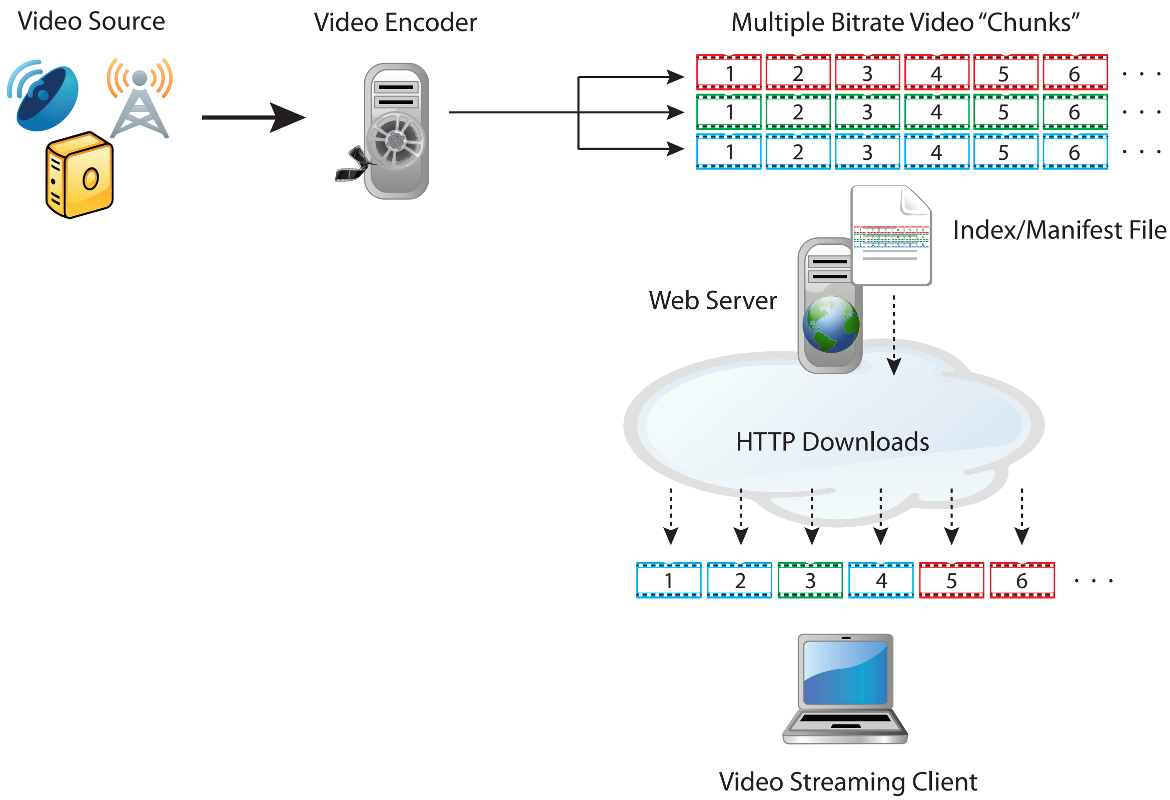
The streaming client demultiplexes the container bitstream into its individual video and audio streams, and the resulting stream packets are stored in a playout buffer before being decoded and played. Some buffering is generally necessary to ensure smooth playback when video is streamed over TCP, which provides reliable delivery of packets but highly variable data rates due to error-checking and retransmissions. The initial buffering may result in a delay before the video starts, with the length of the pause determined by the user's connection speed, the video bitrate/file size and the buffer size. Ideally, the buffer should be small enough to minimize the initial delay, but large enough to ensure that it doesn't completely empty during playback, causing the video to pause while the client rebuffers.

In HTTP-based streaming, the media content is divided into "chunks" that comprise a small portion (typically 2-10 seconds) of video and audio. In some implementations, each chunk is saved as a separate file, while in other cases the content is saved as a single file and each chunk represents a different section of the file. Chunks are typically split at Group of Pictures (GoP) boundaries—each chunk begins with an I frame that is encoded without reference to other frames, making each chunk independent. The chunks are indexed on the server with a manifest or playlist file that directs the client application to the download URL for each chunk. The client sends a series of HTTP requests to download and play each chunk in order; although each chunk may actually be a separate file, the video playback appears seamless to the end user.

The predominant HTTP streaming protocols currently in use support adaptive streaming (see Figure 2), in which the video content is encoded at different bitrates/resolutions and the client player can switch between different quality streams as needed. This allows the streaming client to begin playing video almost immediately at a low bitrate/resolution and then shift to a higher quality stream as the download continues, or temporarily change



■ *Figure 1. Video and Audio Encoding, Encapsulation and Streaming*



■ *Figure 2 Adaptive HTTP Video Streaming*

to a lower quality stream during playback due to reduced bandwidth or other changing network conditions. Each bitrate stream is segmented into chunks and indexed in the playlist/manifest file as a separate profile. By monitoring available resources such as download bandwidth and CPU load, the streaming client can determine which stream profile is most appropriate for current conditions and send a URL request for the next chunk in that profile.

Adaptive HTTP streaming can help minimize initial buffering times and stops/starts during playback, and in most cases can be implemented using a normal web server without the need for streaming server software. In addition, HTTP typically works through firewalls and NAT, while RTMP/RTSP/RTP may not. However, adaptive streaming requires an encoder that is capable of encoding the source video at different bitrates, and can significantly increase storage requirements as multiple "copies" of the same video content must be stored.

Video Streaming Protocols

There are a variety of streaming protocols currently in use, a number of which are proprietary. The following sections provide an overview of commonly used streaming protocols.

Apple HTTP Live Streaming

HTTP Live Streaming (HLS) is a proprietary adaptive bitrate streaming protocol developed by Apple to deliver streamed content to iOS-based devices (iPhone, iPad, AppleTV) and desktop computers (Mac OS X). Encoded video and audio streams (typically H.264 and AAC, respectively) are encapsulated in an MPEG-2 Transport Stream. The MPEG2-TS stream is broken into segments that are saved as media .ts files and indexed with a text-format .m3u8 playlist file. HLS can be used with most web servers including Apache.

Microsoft Smooth Streaming

Like HLS, Smooth Streaming is a proprietary adaptive bitrate streaming protocol that breaks video content into small segments that are delivered using HTTP over TCP. Smooth Streaming encapsulates H.264 or VC-1 video streams and AAC or WMA audio streams in an MP4 container, and uses a smaller segment size (a default of 2-second "chunks," contrasted with HLS's default 10-second chunks). Smooth Streaming requires the use of a Microsoft IIS web server and is primarily used to deliver video content to the Microsoft Silverlight and Xbox Live platforms.

Adobe Flash/HDS

Adobe Flash is a proprietary streaming protocol that is widely used to deliver web video content, such as YouTube, Yahoo! Videos and Hulu, to the Adobe Flash Player or AIR platform. Encoded video and audio streams are encapsulated in an FLV or FV4 container and "pseudo-streamed" as a progressive download using HTTP over TCP, or streamed using RTMP (Real-Time Messaging Protocol) over TCP. A newer Adobe protocol, Adobe HTTP Dynamic Streaming (HDS), is an adaptive streaming protocol that is an alternative to RTMP, and uses HTTP over TCP; like Flash, it can be used to deliver video content to devices compatible with Flash Player or Adobe Air.

DASH

Dynamic Adaptive Streaming over HTTP (DASH), also known as MPEG-DASH, is a non-proprietary streaming protocol similar to Apple HLS, Microsoft Smooth Streaming and Adobe HDS, which was published as an international standard (ISO/IEC 23009-1) in April 2012. Like the proprietary protocols previously mentioned, DASH is an adaptive bitrate streaming protocol that breaks the content into a sequence of small file segments that are downloaded to the client using HTTP. DASH is audio/video codec agnostic and can be used to stream any media data, but most commonly uses the MP4 or MPEG2-TS container format. DASH is currently used to stream some YouTube and Netflix content.

Measuring Video Streaming Quality

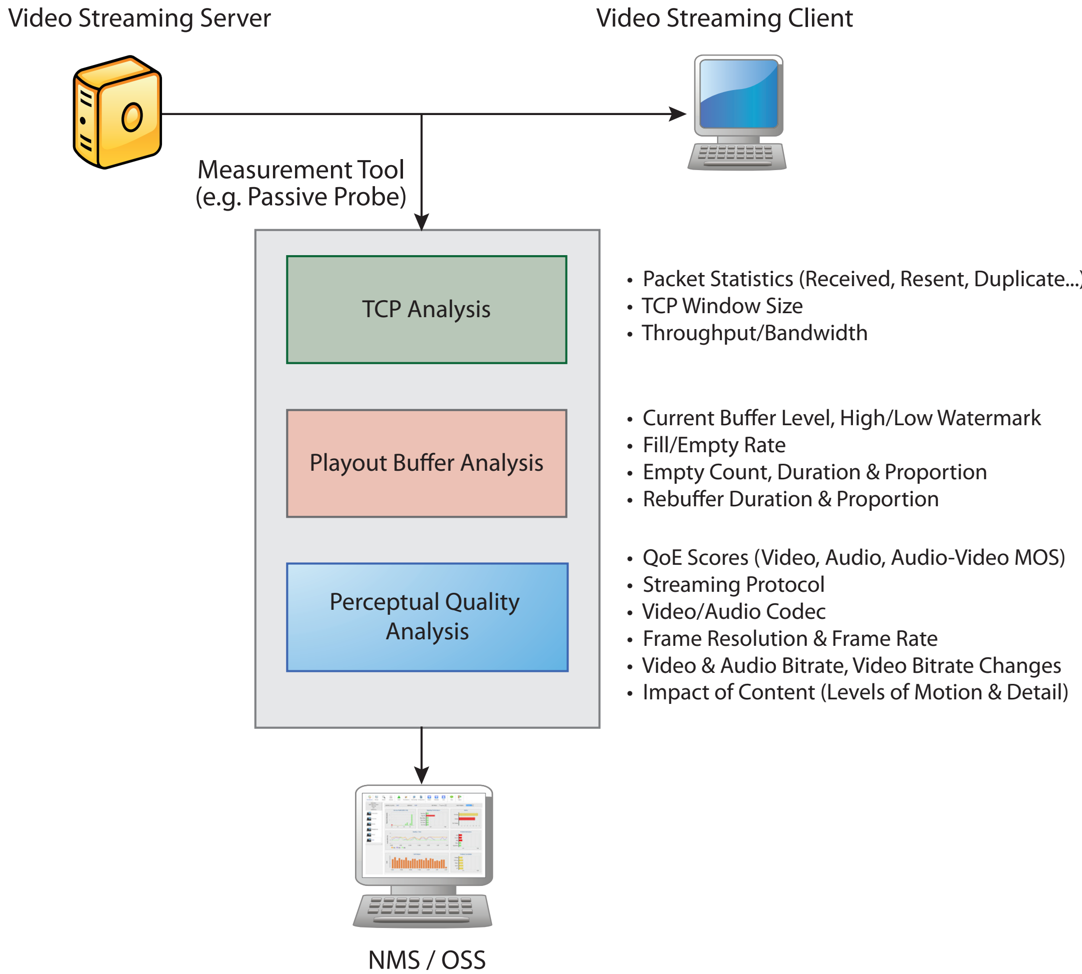
End user perception of video streaming quality is affected by a number of factors, including:

- ***Resolution*** - generally speaking, higher frame resolutions are perceived as better (for example, 4K vs. 640 x 480 SD). However, HD video with a poor frame rate or multiple stops/starts during playback may be perceived as worse by end users than a lower resolution video that plays smoothly.

- ***Frame rate*** - frame rates for streamed video typically vary from around 30 (29.97) frames per second down to 15fps or below; higher frame rates appear smoother, but increase the bandwidth and CPU requirements for the client system. A frame rate of 15fps or below may appear "choppy," but in general a consistent frame rate is preferable to one that varies during playback. A higher frame rate is more essential for high-motion scenes, such as a sporting event, than more static scenes such as footage of a news anchor or an instructional video.

- ***Bitrate*** - video encoded at lower bitrates has a smaller file size and lower bandwidth requirements at the expense of reduced image quality. If adaptive streaming is used, perceptual quality is also impacted by the number of bitrate changes during playback, especially those from a higher to a lower bitrate.

- ***Playback performance*** - this is the end user's perception of the overall "smoothness" of the video playback, impacted by the initial buffering time and the number and length of pauses to rebuffer.



■ *Figure 3. Video Streaming Quality Measurement*

- ***Audio quality*** - the perceptual quality of the accompanying audio stream(s), determined in part by the audio bitrate and audio codec used.
- ***Audio/video synchronization*** - audio that is out of sync with the video may be very annoying, particularly in scenes with dialogue.

All of these factors have an impact on viewer QoE, and must be taken into account when measuring the quality of a streaming video service.

The most widely used video quality metric is the Mean Opinion Score (MOS), a 1-5 measurement of video or audio quality, with 1 being bad/ unacceptable and 5 being excellent. Originally used as a measurement of speech quality in telephony applications, MOS is traditionally obtained using a pool of human subjects who are asked to watch video clips and rate their quality on a scale of 1-5. MOS is calculated as the mean or average of all the subjective scores for each clip, and represents (as closely as possible) a typical end user's perception of video quality. Because using human subjects is obviously not feasible for monitoring the quality of video services in real time, tools such as Telchemy's VQmon® have been developed that use various objective measurements to estimate MOS values correlating to those obtained from subjective survey.

MOS metrics for video streaming include Video MOS (MOS-V), Audio MOS (MOS-A), and Audio-Video MOS (MOS-AV), which provides an overall QoE measurement of audio and video quality and audio/video synchronization. MOS is impacted in part by the video or audio codec in use; because all codecs introduce some degree of distortion, in practice a perfect MOS of 5 is not attainable.

Figure 3 depicts a set of quality and performance metrics for a typical video streaming service, obtained using a measurement tool (in this case, a passive probe with video and audio analysis capabilities).

Factors Affecting Quality Measurement

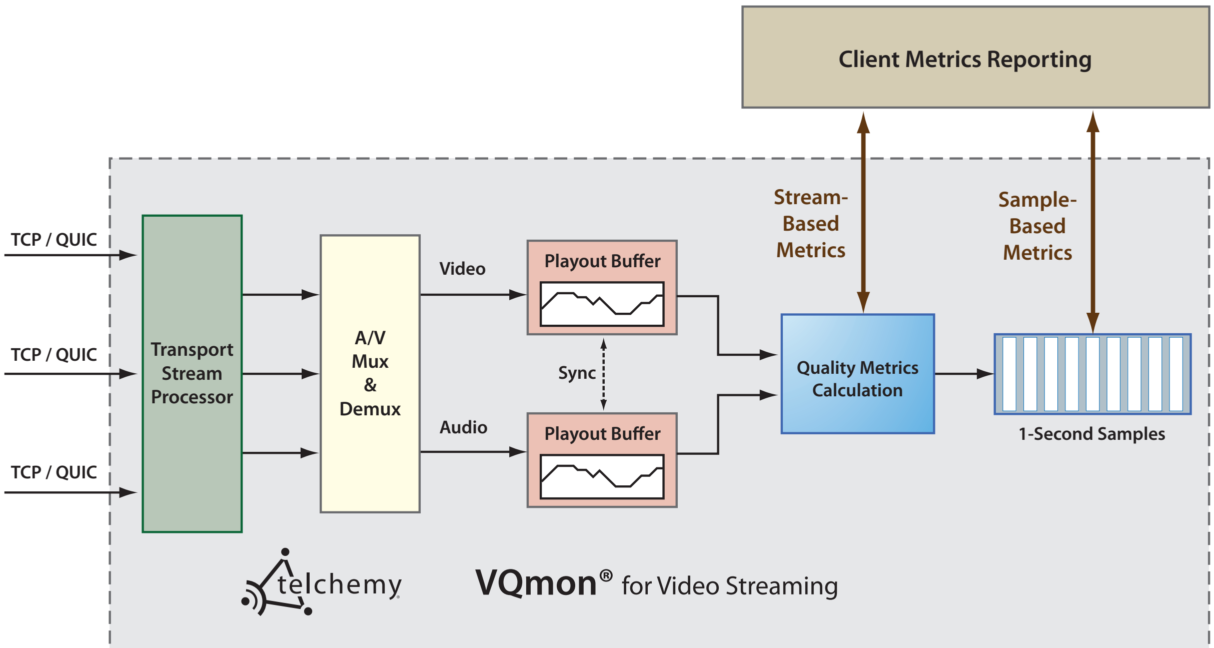
A number of factors can impact the ability to accurately measure the quality of streamed video. These include:

- ***Encryption*** - the use of encryption in video streaming, for purposes of security and/or digital rights management, is increasingly common and presents an obstacle to some methods of quality measurement. Encryption can obscure many details about the video and audio stream—including the codec type, frame resolution, bitrate and other characteristics—that are used to estimate MOS. Limited information, including TCP transport metrics and statistics for file segment downloads, can still be obtained in many cases, but accurate perceptual quality calculation is difficult without access to the decoded streams.
- ***Parallel TCP Connections*** - some video streaming services allow the client player to request file segments over multiple TCP connections to increase throughput. In some cases this can result in file segments arriving out of order. The quality measurement method must have the ability to recognize and reorder out-of-sequence file segments before they are processed.
- ***Proprietary Codecs and Streaming Protocols*** - many video and audio codecs and streaming protocols are either partially or wholly proprietary in nature, and new codecs are continually being developed. Moreover, streaming behavior is largely influenced by the type and configuration of the streaming client. The wide range of possible configurations and lack of a standard delivery format make quality measurement considerably more complex—in part, this problem may be alleviated in the future by industry adoption of MPEG-DASH and other developing standards.

VQmon for Video Streaming

VQmon for Video Streaming is an embedded software agent that passively monitors the quality of streamed video sessions in real time. As part of Telchemy's VQmon family of modular performance monitoring software products for IP video and Voice over IP, VQmon for Video Streaming is specially designed to monitor and analyze the performance of video and audio streamed over TCP, and supports most common streaming protocols, container formats, and video and audio codecs. VQmon is a compact, non-intrusive technology that can be integrated directly into routers, gateways, probes, handheld test equipment, and other network infrastructure and client devices.

VQmon for Video Streaming uses an advanced perceptual quality algorithm to produce real-time QoE scores (MOS-V, MOS-A, MOS-AV) and a detailed set of performance metrics for streaming video sessions. The VQmon algorithm calculates Video MOS frame by frame, while considering multiple factors including codec type, actual bitrate, frame resolution and frame rate, and scene content characteristics including levels of motion and detail.



■ *Figure 4. VQmon for Video Streaming - Real-Time Quality Analysis and Metrics Reporting*

Figure 4 is a functional diagram depicting the processing flow of VQmon for Video Streaming. Incoming TCP streams are processed by VQmon's TCP stream processing module. (In the case of multiple parallel TCP downloads to the same client, VQmon can detect and process file segments that are received out of sequence.) The stream processing module extracts detailed transport-related metrics as it passes the reassembled media stream(s) on to the audio/video mux & demux module. This module separates each multiplexed stream into separate video and audio streams, which are then processed through separate video and audio playout buffer emulators to extract buffer statistics and information on audio-video synchronization. VQmon's quality metrics algorithm analyzes the input data and produces perceptual quality scores (including MOS) and additional metrics, including detailed measurements of playout buffer conditions, at a sample interval of one second. These sample-based metrics, along with a set of stream-based TCP metrics, are made available to external client applications via function calls to VQmon's API.

VQmon for Video Streaming supports many popular streaming protocols including Apple HLS, Microsoft Smooth Streaming, and Adobe RTMP/Flash 9+/HDS, along with a range of container formats, video and audio codecs. VQmon may not be able to decode encrypted video streams, but may report some information about downloaded file segments when encryption is in use.

telchemy®

Summary

Video streaming is an increasingly popular method of delivering various types of video content to end users. Tools such as Telchemy's VQmon use objective measurements to provide real-time feedback on the subjective viewer experience.

About Telchemy, Incorporated

Telchemy® is the global leader in analytics technology for real-time applications and multimedia IoT with its VQmon®, Embiot®, DVQattest®, SQprobe® and SQmediator® families of service quality monitoring and analysis products. Telchemy pioneered the use of embedded analytics technology and the application of big data for VoIP and Video performance management, and is positioned to be a leading provider of analytics technology for the emerging IoT market. Founded in 1999, the company has products deployed worldwide and markets its technology both directly and through many leading networking, test and management product companies.

For more information, please visit **www.telchemy.com**.

Download application notes:
www.telchemy.com/application.html

Telchemy Application Notes

Series IP Video Performance Management

Title Measuring the Quality of Video Streaming Services

Title IP Video Encoding Explained

Title Measuring IP Video Quality

Title Understanding Video Quality Metrics

Title Managing IPTV Performance

Information

Email
sales@telchemy.com
info@telchemy.com

Phone
+1-678-387-3000

Telchemy Incorporated
105 Nobel Ct
Suite 100
Alpharetta, GA 30005
USA