A STATISTICAL-ANALYTICS APPROACH TO MAKING VIDEO BANDWIDTH AND QOE DECISIONS WITH CONFIDENCE

Sean T. McCarthy, Ph.D.
ARRIS

Abstract

This paper presents a new statistical-analytic approach intended to enable operators to make video bandwidth and QoE decisions with confidence. The method we present is based on a new way of describing video-quality and bandwidth efficiency in terms of statistical probabilities. It can be applied to any video distribution method including ABR, CBR, and statmux for any format. A significant aspect of our method is that is does not require explicit traditional measurement of video quality in terms of PSNR, SSIM, or MOS values. Instead, we show that a metric derived from program complexity can be used as a statistical indicator of quality. Finally, this paper will show how real world data from in-service operations can be used to address key performance questions such as: What is the probability that video quality drops below any given level? Which programs are not receiving enough bandwidth? What is the efficiency of dynamic bandwidth allocation (VBR or ABR) compared to CBR allocation to each video program? Which operational parameters could be changed to improve overall video quality and efficiency? How would introduction of a new service impact existing services?

INTRODUCTION

Television has become more diverse and complex; but one thing remains the same -- operators want to deliver great video experiences to subscribers with optimal use of bandwidth. Yet, as the number of formats, displays, codecs, distribution protocols, and access technologies proliferate, it becomes increasingly more difficult for an operator to know with confidence that performance targets are being met. There are simply too many permutations to test using traditional methods. The challenge is compounded when new technologies such as DOCSIC 3.1 and HEVC enable 10's or even 100's of individual video programs to share bandwidth.

The motivation of the work reported in this paper was to find a way of assessing the performance of video distribution without needing to measure video quality for each individual program directly using methods such as PSNR, SSIM, JND, or MOS subjective testing. None of those tried-and-true methods scale very well in a mixed world full of adaptive bit rate protocols (ABR), CBR, statistical multiplexing, and a variety of core codecs (MPEG-2, AVC, HEVC, etc.) and numerous distribution models (IPTV, DTH, OTT, QAM, DOCSIS, etc.)

The new alternative method we have developed and have been exploring is based on the concept of "video-quality stress," which we define as the moment-to-moment ratio of 1) the varying bitrate that would be needed to achieve a target constant video quality and 2) the bitrate that is actually allocated to a program.

Moreover, we have found it useful to think of video-quality stress in terms of probability distributions rather than as a single value. The probability distributions provide insights into how stress changes frame-to-frame, scene-to-scene, and across different programs. Thinking in terms of probabilities opens up new ways of setting operational performance targets.

For example, instead of setting a bitrate to achieve or exceed a particular MOS value, the use of probability distributions allows a more nuanced approach in which bitrates can be allocated so that programs meet or exceed a particular video-quality stress target for a predetermined percentage of time. In other words, the use of video-quality stress probabilities can enable an operator to set expected video-quality performance targets with foreknowledge of how often actual programming might deviate from the target. We believe that both the target video-quality stress and the probability of deviation are useful components in understanding a consumer's overall quality of experience (QoE), and could thus prove to be very useful to operators when designing video programming product offerings.

VIDEO-QUALITY STRESS

We define "video-quality stress" as the ratio of the bitrate that would be needed to produce constant video quality to the actual bitrate of the video program.

Video-quality stress is illustrated in Figure 1 for three video-distribution use cases: constant bitrate encoding (CBR); variable bitrate statistical multiplexing; and adaptive bitrate (ABR).

Figure 1 panel ***A*** illustrates the hypothetical variable bitrate (VBR) that would be required to maintain constant video quality over time for a particular program. We call this hypothetical variable bitrate a "need parameter." The need parameter is thus a measure of the moment-to-moment complexity of the particular video program with respect to compression. (The manner by which need parameter values may be obtained is discussed below.)

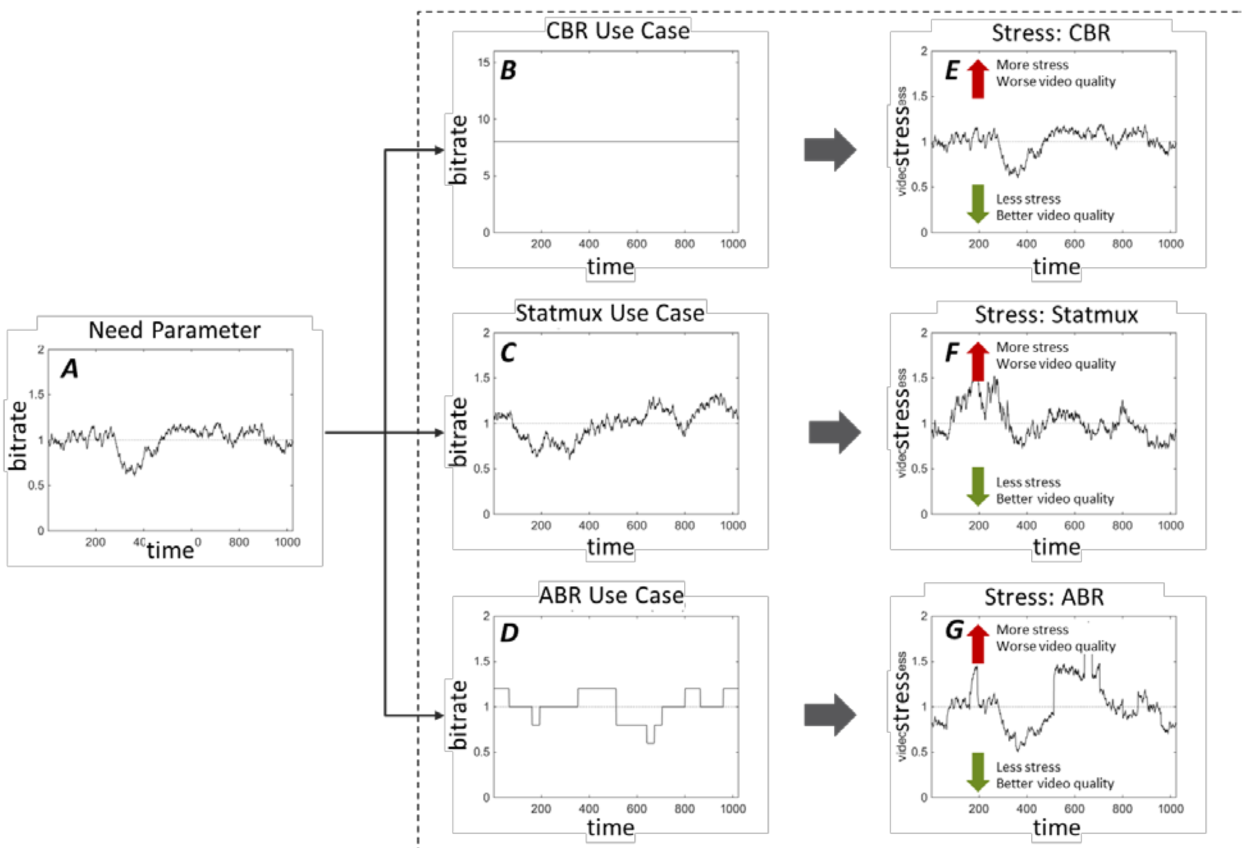

Figure 1 – Illustration of video quality stress

CBR encoding of a video program having the need parameter shown (***A***) is illustrated in Figure 1 panel ***B***. In the example shown, the CBR bitrate is equal to the average hypothetical VBR bitrate (the average value of the need parameter).

An example of the same video program being encoded as part of a statistical multiplexed (statmux) group of programs is illustrated in Figure 1 panel ***C***. In statmux, several programs are grouped into a pool that shares a typically fixed total bandwidth (see Figure 2). The objective of statistical multiplexing is to equalize video-quality across programs. This is achieved by dynamically allocating a portion of the total bandwidth to each individual program based on its complexity (as represented by a "need parameter", for example) compared to the complexity of the other programs in the pool. When a program has a high relative complexity the statmux controller will allocate it a greater share of total bandwidth, and, conversely, a lesser share when its relative complexity is low. The result is a variable-bitrate video bitstream as illustrated in panel C. (Note that only one program from the statmux pool is shown.)

Finally, an adaptive bitrate (ABR) use case is illustrated in Figure 1 panel ***D***. In the example shown, the video program is delivered as a series of contiguous piecewise-continuous CBR segments. The bitrate of each segment is allowed to change from segment to segment. The bitrate changes are typically driven by requests from client devices in response to local bandwidth availability.

The video-quality stress corresponding to each use case is shown in Figure 1 panels ***E*** (CBR), ***F*** (statmux), and ***G*** (ABR). In the simple CBR example shown in Figure 1***E***, the video-quality stress is the need parameter value, ***A***, divided by the CBR bitrate, ***B***, for each point in time. Similarly, the video-quality stress for the statmux example is mathematically the need parameter value divided by the actual statmux bitrate, ***C***. Video-quality stress for the ABR is obtained in the same manner for the data illustrated in ***A*** and ***G***.

Video-quality stress is not an absolute measure of video quality in the same way as PSNR, SSIM, JND, and other traditional video-quality metrics. Rather, video-quality stress is an indicator of the deviation of quality from a target.

In the CBR example, illustrated in Figure 1, ***B*** and ***E***, the CBR bitrate is deliberately set equal to the average value of hypothetical VBR bitrate (the need parameter) to make a point. One should expect that about half the time the CBR video quality would be better than the hypothetical VBR video quality because about half the time the hypothetical VBR bitrate – the bitrate needed for constant quality – is less than the average VBR rate (the CBR rate). Likewise, about half the time, CBR video quality would be worse than that for VBR.

Video-quality stress is thus an indicator of the deviation of video-quality from a predetermined benchmark. A value of video-quality stress equal to 1 indicates that a program is matching the video-quality benchmark. A value greater than 1 indicates that the program is stressed and video-quality is likely worse than the benchmark. A video-quality stress value less than 1 indicates that the program is less stressed and video quality is likely to be greater than the benchmark. (Below, we describe a method of calibrating video-quality stress to the video-quality benchmarks chosen by a video-service provider.)

The variation in video-quality stress for the statmux and ABR use cases are illustrated in Figure 1 ***F*** and ***G***, respectively. In each case, the dashed horizontal line corresponds to a video-quality stress factor equal to 1. Deviations above the line indicate better-than-benchmark video quality, and deviations below indicate worse-than-benchmark video quality.

Video-Quality Stress to Validate Performance

Figure 3 illustrates the use of video-quality stress to validate the performance of a statmux system. In this example, 3 commercial video programs were fed into a 3-program statmux system in our lab, as illustrated in Figure 2. The system was made up of 3 encoding modules and 1 controller module that managed the allocation of bandwidth between the programs by sending bitrate allocation messages to each encoding module. In addition to producing a compressed program bitstream, each encoding module also produced a need parameter time series for the corresponding input program.

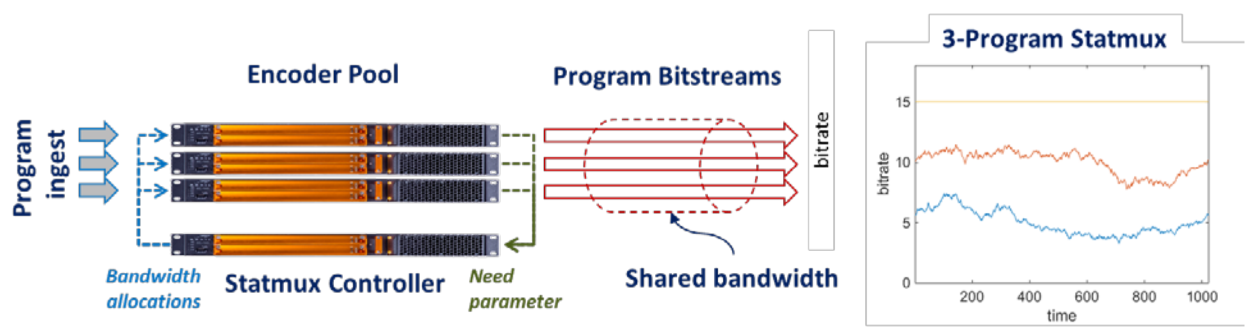

Figure 2 – Statistical Multiplexing

In Figure 3, examples of the resulting statmuxed video bitrates (***A***, ***D***, and ***G***) and the corresponding need parameter time series (***B***, ***E***, and ***H***) are shown. The corresponding video-quality stress for each program is shown individually in ***C***, ***F*****,** and ***I***. The video-quality stress time series are also plotted together in ***J***. Note that the video-quality stress is very similar for all programs thus indicating that the statmux system is correctly doing its job of equalizing video quality across the programs.

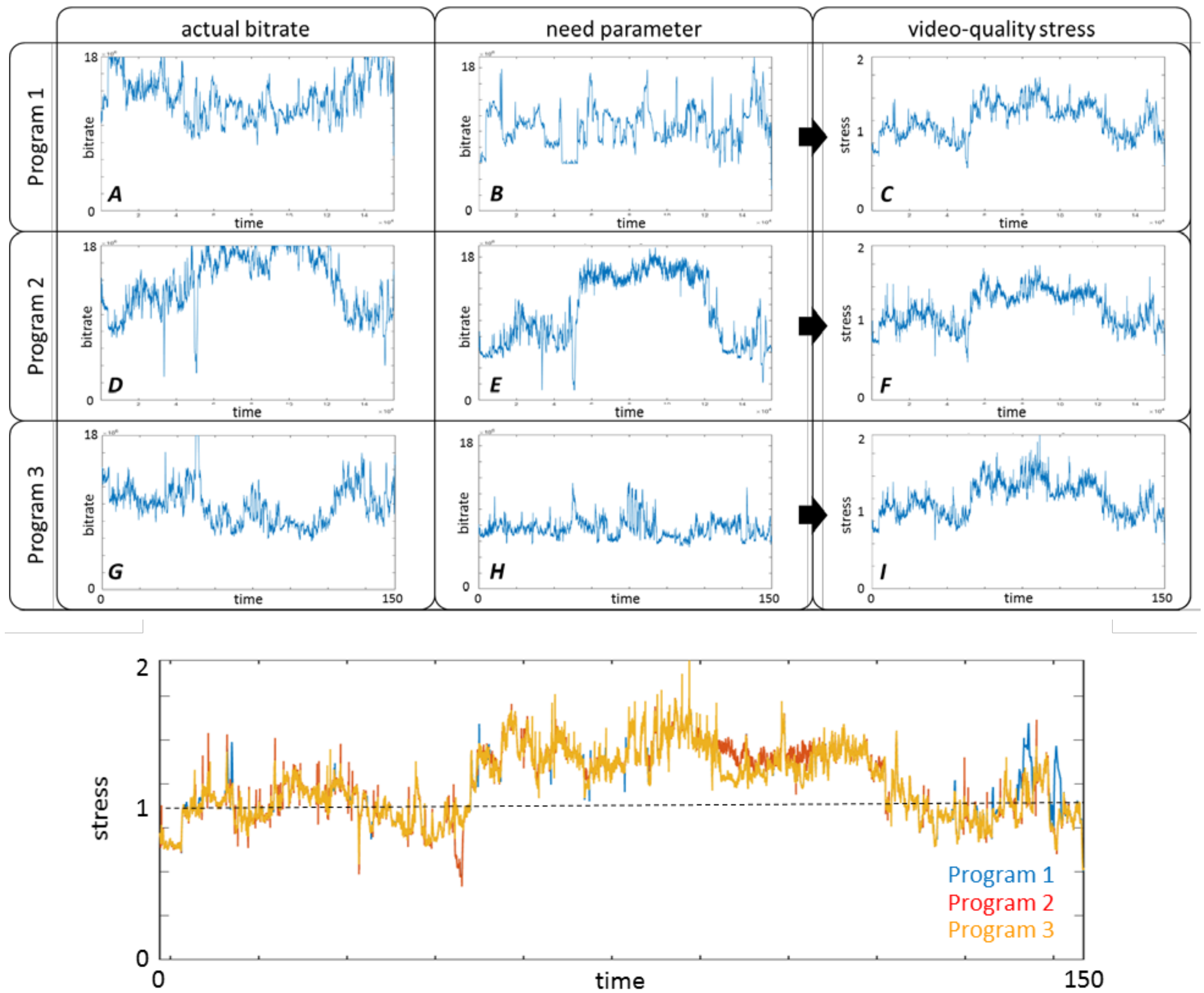

Figure 3 – Illustration of Video-Quality Stress Equalization in Statistical Multiplexing

<u>Generating Need Parameters</u>

It is worth noting that over the years we have developed and optimized algorithms for calculating need parameters for common video formats and codecs. Most of this work came from our work on statistical multiplexing where our approach has been to analyze video as part of the encoding process and thereby generate a need parameter that is provided to statmux controllers which then allocates bandwidth accordingly. Though born of statmux, need parameters can be calculated separate from the encoding process for use in any application (ABR, CBR, etc.). Other vendors working in this field have developed similar metrics, though perhaps under different names. What is important for the purpose of this paper is that the need parameter be as proportional as possible to the hypothetical VBR bitrate that would be needed to achieve constant video quality.

<u>Calibrating Video-Quality Stress</u>

Video-quality stress is a relative rather than absolute metric. It is a quantified indicator of video quality relative to an implicit video quality benchmark.

Setting video quality performance targets is something that video providers already do.

Indeed, setting appropriate video quality targets is a critical part of setting overall QoE targets that drive successful commercial services. The absolute video quality (as measured by subjective testing, for example) of a successful service depends on many factors, and can vary between free and premium services.

Once video-quality benchmarks are selected for a particular service, that absolute video quality level can be used to calculate a *video-quality-stress calibration factor* using steps equivalent to the following:

1. Collect bitrate data for a set of programs and times that is representative of a service offering.

2. Collect need parameter data for the same set of programs and times as in step 1.

3. Calculate the average value of the bitrate data collected in step 1.

4. Calculate the average value of the need parameter data collected in step 2.

5. Calculate a *video-quality-stress calibration factor* by dividing the average value of the need parameter data by the average value of the bitrate data.

With a *video-quality-stress calibration factor* in hand, a video provider can determine video-quality stress on an on-going basis. Calibrated video-quality stress is calculated by multiplying any need parameter time series by the *video-quality-stress calibration factor* and then dividing by the corresponding bitrate time series. When calibrated correctly, the expectation of the video provider should be that video-quality stress thus determined should fluctuate around 1 on an on-going basis; i.e., the average video quality of the service should be the equivalent to the video quality of the sample data set used in calibration.

Comparing Services

It is worth noting that a video provider could calibrate a premium flagship service, for example, and then use the resulting *video-quality-stress calibration factor* to gain insight into the relative quality of other services. For example, a video provider could use the *video-quality-stress calibration factor* obtained for the premium service as the calibration factor for other services. For example, this technique could be used to gage the relative quality of a new service under consideration, or to quantify the relative quality of premium services compared to non-premium services.

VIDEO-QUALITY STRESS PROBABILITY

The notion of video-quality stress by itself could lead to new insights; but we believe its real power comes when it is considered in terms of probabilities.

In Figure 4***A***, we show video-quality stress for a dozen simulated programs. (The simulated set of programs is shown as an illustration on an operator collecting a sample of programs from actual operations or by use of a standard video library used for internal testing and planning.) In the example shown, each program has a video-quality stress that tends to deviate from the calibration point (1, in this case) but by different amounts.

The ensemble likelihood that video-quality stress would exceed a particular value can be described with a cumulative probability, as shown in Figure 4***B***.

Cumulative probability provides a quantitative basis for answering questions such as, “how often does video-quality stress exceed a certain value?” For example, the lower-left crosshair in Figure 4B indicates that approximately 20% of the time video-quality stress would be less than ~0.8. The upper-right crosshair indicates that video-quality stress would be more than ~1.4 only approximately 10% of the time.

Recall that video-quality stress is the ratio of a hypothetical bitrate needed to achieve a predetermined video-quality benchmark relative to the actual bitrate. Thus, a video-quality stress of 0.8 corresponds to a situation in which 125% of the hypothetical constant-quality bitrate is actually being consumed by a program. The lower-left crosshair in Figure 4B therefore indicates that more than 125% of the benchmark content-quality bitrate is being used by programs approximately 20% of the time.

Similarly, the upper-right crosshair indicates that approximately 10% of the time programs are getting less than ~70% (1/1.4) of the bitrate they would need to achieve the benchmark constant-quality target.

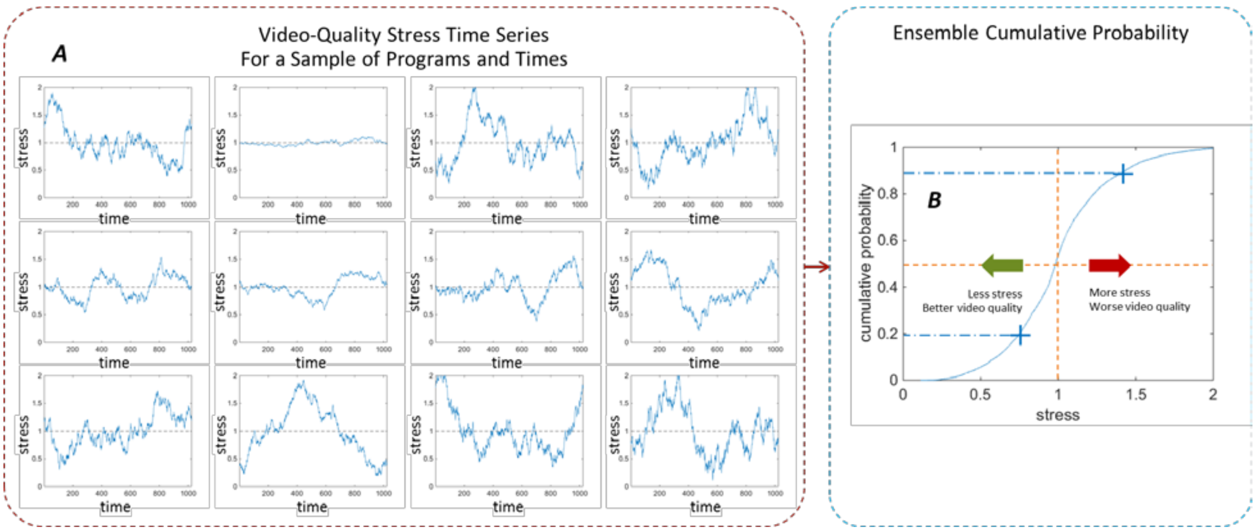

Figure 4 – Illustration of a determining Video-Quality Stress Cumulative Probability

Tuning Video-Quality Performance

In Figure 5, we show how video-quality stress can be used to fine tune video-quality performance. The key idea is that a service operator can determine how much bandwidth to allocate to a particular program in order to achieve performance defined by how often a program's video quality is likely to meet or exceed a benchmark video-quality target.

For the example shown in Figure 5, the cumulative probability of video-quality stress for a program (or group of programs) is shown as the right-most curve. Allocating more bandwidth to a program (or group of programs) would have the effect of moving the cumulative probability to the left. The amount of shift desired quantifies the amount of bandwidth that should be allocated.

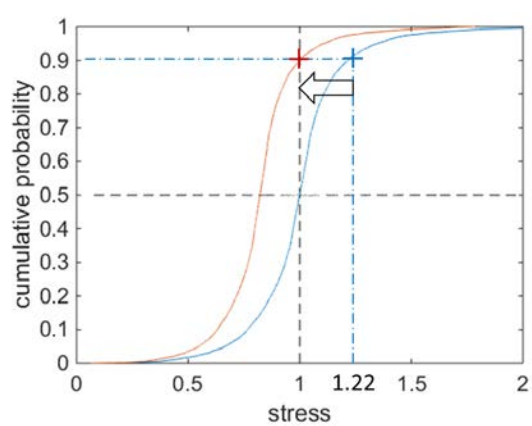

Figure 5 – Calculating Bandwidth to Achieve Video Quality Performance Targets

If for example an operator wished to provide a program having video quality equal or better than a benchmark target for 90% of the time, the operator could take the following steps:

1) Calculate the cumulative probability for an original bandwidth.

2) Find a first video-quality stress along the horizontal axis that corresponds to 50% along the vertical axis. (In this example, that video-quality stress is 1).

3) Find a second video-quality stress along the horizontal axis that corresponds to 90% (in this example) along the vertical axis.

4) Divide the second video-quality stress value by the first video-quality stress value. (The result is 1.22 in the current example.)

5) Multiply the result into the original bandwidth. The result is the bandwidth that if it were allocated to the program would achieve the desired performance target. (The tuned bandwidth would be 122% of the original bandwidth in this example.)

It is worth noting that an operator could use the method described above to explore several candidate performance targets to arrive and the best balance of video-quality and bandwidth utilization. Such steps can be performed for individual program or for any group of programs or even an operator's entire programming line-up.

<u>Bandwidth Sharing</u>

Video-quality stress can be calculated for any group of channels sharing bandwidth, as illustrated in Figure 6 for a statmux use case. The moment-by-moment sum of the need parameters for each program in the statmux pool yields an aggregate need parameter, which in turn yields an aggregate video-quality stress for the entire pool of programs. The same principles could be applied to gain insight into the aggregate performance of managed and unmanaged ABR distribution.

Statistically, the total accumulated neediness of the programs varies over time. Consequently, the total video-quality stress varies over time. The amount of that variability depends on the number of programs sharing bandwidth.

Figure 6 illustrates the dependence of aggregate video-quality stress on program count for a statmux use case. The left-hand graphs represent statmux pools of 3, 30, and 300 programs, respectively. The middle graphs illustrate the corresponding aggregate video-quality stress over time. Note that the variability of aggregate video-quality stress decreases with increasing program density (from top to bottom in the figure). The rightmost graph shows that the standard deviation of the aggregate video-quality stress decreases in inverse proportion to the square-root of the number of programs in the statmux pool.

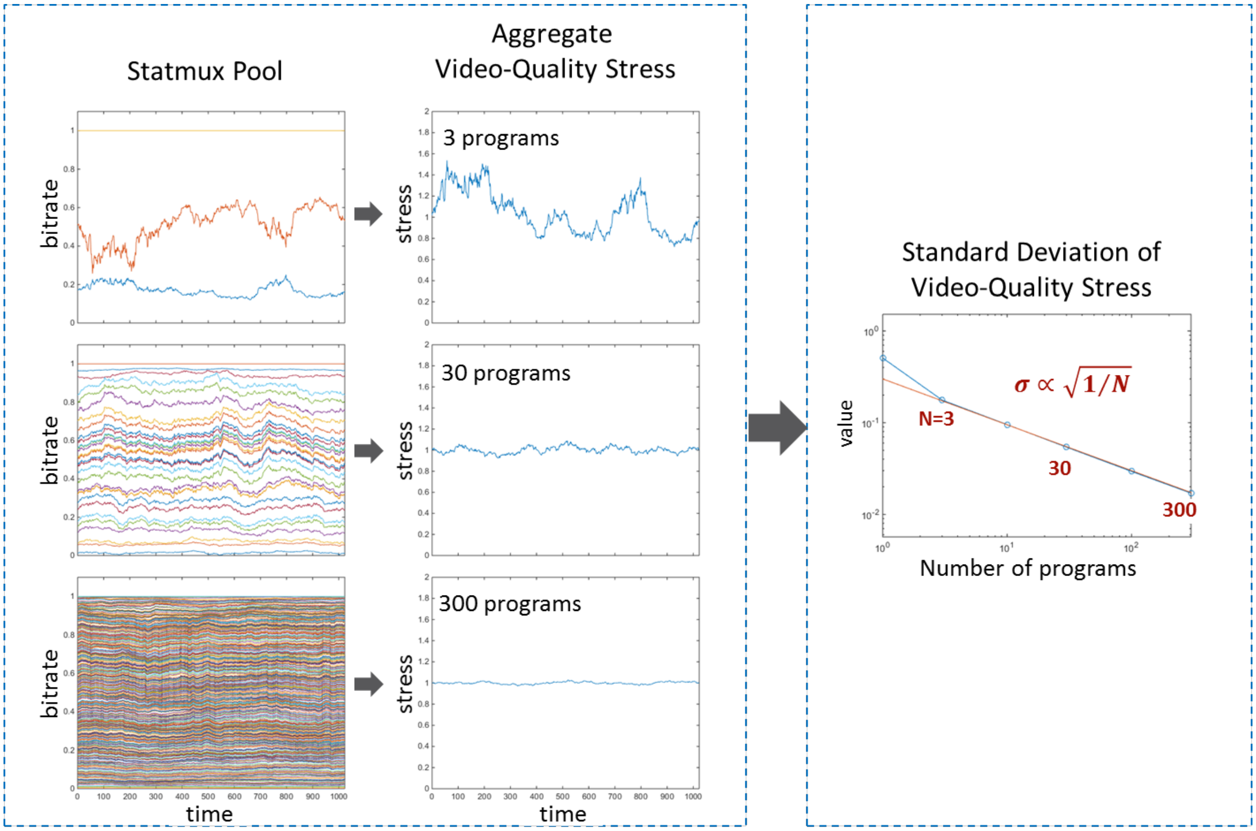

Figure 6 – The variability of the video-quality stress decreases in a predictable manner with program density

Another way to visualize the impact of program density on variability is illustrated in Figure 7. Panel ***A*** shows cumulative probability distributions for statmux systems having 1, 3, 10, 30, 100, and 300 programs, respectively. As the number of programs in the statmux pool increases, the cumulative probability curve narrows and becomes steeper. Panels ***B*** though ***G*** are the statmux equivalents of the shifted cumulative probability curves that were shown in Figure 4.

As the number of programs in the statmux pool increases, the amount by which the cumulative probability curve needs to shift decreases. In other words, operators having a large number of programs sharing bandwidth might find opportunities to optimize video-quality performance with less investment in additional overall bandwidth per program.

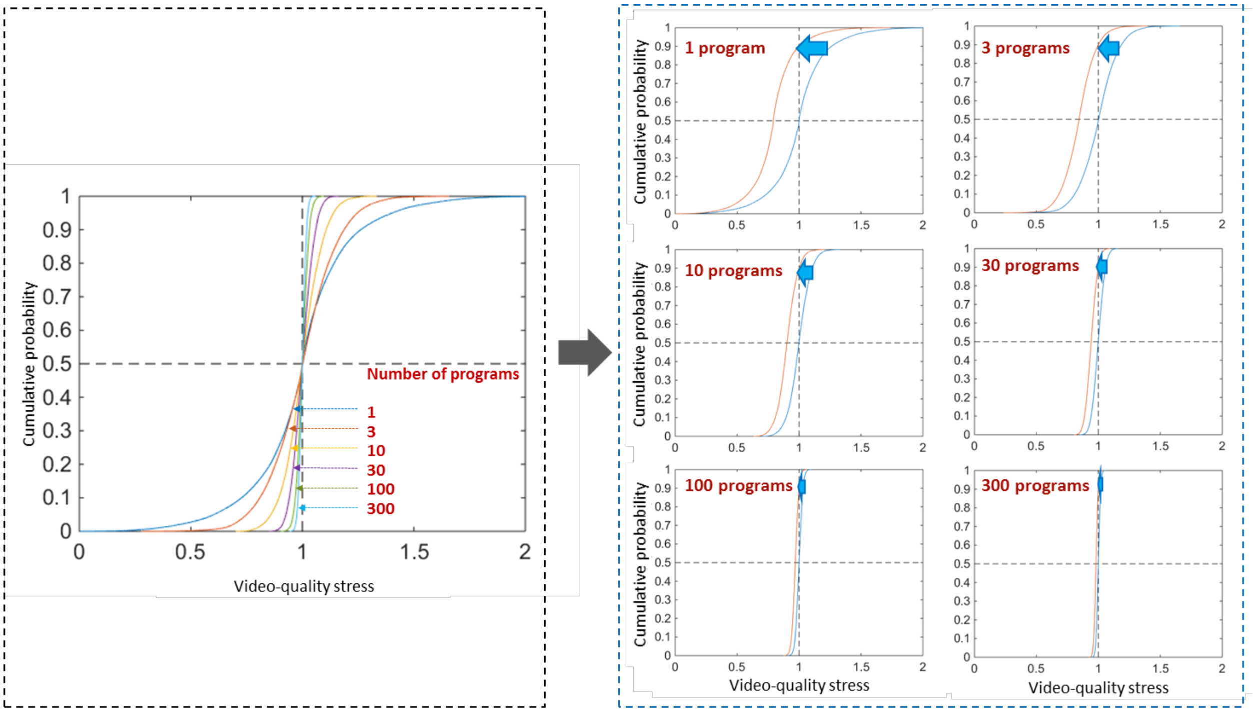

Figure 7 – The cumulative probability distribution of the video-quality stress narrows and steepens with increaseing program density.

CONCLUSION

The objective of the work reported in this paper was to find a way of assessing the performance of video distribution without needing to measure video quality of each individual program directly using methods such as PSNR, SSIM, JND, or MOS subjective testing. Such an alternative would provide advantages of scalability that are not otherwise possible, which is of particular importance as the number of programs increase and the ways in which those programs are distributed proliferate.

A key concept we introduced and explored in this paper is "video-quality stress," which we define as a relative indicator of video quality with respect to a video-quality benchmark. Video-quality stress is derived from a "need parameter" and the actual bandwidth consumed by a video program. The need parameter is a metric that is extracted from decades of statistical multiplexing experiences, and is often available as metadata (or closely related metadata) from professional encoders and transcoders. For the purposes of this paper, the need parameter is a moment-by-moment indicator of the hypothetical variable bitrate that would be needed to achieve constant video quality over time. In this sense, the need parameter is a measure of the complexity of a program.

We provided a method that operators can use to calibrate video-quality stress to the particular needs of their own video services. We also showed how a *video-quality stress calibration factor* could be used to compare the relative video quality of difference services.

Another key concept presented is the notion of describing video-quality stress in terms of probabilities. Doing so enables more nuanced consideration of video-quality performance and benchmarks. Calculation of cumulative probabilities from programs and/or groups of programs enables operators to gain insight into how often the video quality of individual program or groups of programs exceed or drop below any particular level. Perhaps as important, we showed how video-quality stress probability could be used to tune bandwidth allocation to programs so as to achieve quantifiable performance targets.

Lastly, we explored the behavior of video-quality stress when a large number of programs share bandwidth and were able to quantify the dependence of the variability of video-quality stress on program count.

We plan to continue to investigate areas in which the notion of video-quality stress and associated probabilities could provide new actionable insights. In the meantime, we hope that the information we present here will prove useful to operators as they explore the optimal balance between video quality and bandwidth utilization. We welcome the opportunity to explore these concepts more fully with our colleagues and partners in the industry.