# Metrics for Evaluating Video Streaming Quality in Lossy IEEE 802.11 Wireless Networks

An (Jack) Chan*, Kai Zeng*, Prasant Mohapatra*, Sung-Ju Lee† and Sujata Banerjee†
*Department of Computer Science, University of California, Davis, CA 95616
Email: {anch,kaizeng,pmohapatra}@ucdavis.edu
†Multimedia Communications & Networking Lab, Hewlett-Packard Labs, Palo Alto, CA 94304
Email: {sjlee,sujata.banerjee}@hp.com

*Abstract*—**Peak Signal-to-Noise Ratio (PSNR) is the simplest and the most widely used video quality evaluation methodology. However, traditional PSNR calculations do not take the packet loss into account. This shortcoming, which is amplified in wireless networks, contributes to the inaccuracy in evaluating video streaming quality in wireless communications. Such inaccuracy in PSNR calculations adversely affects the development of video communications in wireless networks. This paper proposes a novel video quality evaluation methodology. As it not only considers the PSNR of a video, but also with *modifications* to handle the packet loss issue, we name this evaluation method MPSNR. MPSNR rectifies the inaccuracies in traditional PSNR computation, and helps us to approximate subjective video quality, Mean Opinion Score (MOS), more accurately. Using PSNR values calculated from MPSNR and simple network measurements, we apply linear regression techniques to derive two specific objective video quality metrics, *PSNR-based Objective MOS (POMOS)* and *Rates-based Objective MOS (ROMOS)*. Through extensive experiments and human subjective tests, we show that the two metrics demonstrate high correlation with MOS. *POMOS* takes the averaged PSNR value of a video calculated from MPSNR as the only input. Despite its simplicity, it has a Pearson correlation of 0.8664 with the MOS. By adding a few other simple network measurements, such as the proportion of distorted frames in a video, *ROMOS* achieves an even higher Pearson correlation (0.9350) with the MOS. Compared with the PSNR metric from the traditional PSNR calculations, our metrics evaluate video streaming quality in wireless networks with a much higher accuracy while retaining the simplicity of PSNR calculation.**

## I. Introduction

Multimedia streaming is becoming one of the most popular applications in today's computer networks. Video streaming penetrates every aspect of our lives, ranging from communications to entertainment. With the wide deployment of IEEE 802.11 Wireless Local Area Networks (WLANs), video streaming over WLANs is very common. Video quality measurement, based on criteria and metrics that can be measured objectively and automatically by a computer program, is important to various parties, including government and industries. People evaluate video quality for specification of system performance requirements, comparison of competing service offerings, network maintenance and so on. From the beginning of digital imagery and video, the video research community has proposed a number of metrics to measure video quality. The common metrics include Peak Signal-to-Noise Ratio (PSNR), Structure Similarity (SSIM) index [1], Czekanowski Distance (CZD) [2], etc. PSNR as well as the other objective video quality metrics do not perfectly correlate to perceived visual quality. In addition to the non-linearity of the human visual system, these metrics fail to capture the packet loss characteristics of wireless networks. While these metrics work well for evaluating video quality in the encoding/decoding process and streaming over wired networks, noticeable inaccuracy arises when they evaluate video quality over wireless networks, particularly in lossy networks such as multihop wireless mesh networks. For instance, it could happen that a video stream with a PSNR around 38dB (the full score of PSNR is 100dB) is actually perceived to have the same quality as the original undistorted video. In our subjective video quality evaluation, that will be discussed in Section V-A, all the viewers rate this video stream at the highest subjective quality.

Video streaming applications use UDP, which unlike TCP, provides unreliable transmissions as the transport layer protocol as a trade off for satisfying delay requirements. In WLANs, due to the instability of wireless channels, the probability of a packet loss is much higher than that in wired networks. Losing consecutive *packets* causes the loss of an entire image *frame* in the video's raw format (for example, raw YUV-formatted video file is a sequence of image frames in YUV color space). Most of the objective video quality metrics, including PSNR, are per-pixel quality metrics. They compare every pixel in each frame of a processed video (for example, a video after streaming) with the corresponding pixel in each corresponding frame of a reference video (the original video) to evaluate the quality of a processed video. If a frame in the processed video is lost during streaming, the metrics compare two non-corresponding frames from the processed video and the reference video. This discrepancy results in inaccuracies in the final metric value. We will explain this phenomenon in more detail in Section III-A.

In this paper, we propose a new objective video quality evaluation methodology particularly well suited for video streaming over lossy wireless networks. Because of the popularity and simplicity of PSNR, our evaluation method also

This research was supported in part by the National Science Foundation through the grant CNS-0831914 and the Army Research Office through a MURI grant W911NF-07-1-0318.

calculates the PSNR of a video. However, we modify the traditional PSNR calculation for video so that it handles video frame losses. As it involves the *modification* of PSNR calculations, we name our new evaluation method MPSNR. Using linear regression against Mean Opinion Score (MOS) collected from human subjective evaluation, we derive two specific objective video quality metrics from MPSNR. The first metric, called *PSNR-based Objective MOS (POMOS)*, takes the averaged PSNR calculated from MPSNR as the only input for predicting MOS. Despite its simplicity, it has a Pearson correlation [3] of 0.8664 with the MOS. By adding a few other simple network measurements, such as the distorted frame rate and frame loss rate in a video streaming, the second metric, called *Rates-based Objective MOS (ROMOS*, achieves an even higher Pearson correlation of 0.9350 with the MOS. Using MPSNR, the required parameters, such as PSNR and frame loss rate, can all be measured when both the processed and the reference videos are available. [1] Other objective video quality metrics that closely approximate MOS, such as Perceptual Evaluation of Video Quality (PEVQ) [4] and National Telecommunication and Information Administration Video Quality Metric (NTIA VQM) [5], are complex and do not explicitly handle frame losses in wireless channels. In contrast to these metrics, the proposed MPSNR-based metrics consider frame losses while retaining the simplicity of PSNR.

The contribution of this paper is two-fold:

- We identify the detrimental impact of packet losses during video streaming on video quality metrics, such as PSNR.
- We propose a simple objective video quality evaluation methodology, MPSNR, that alleviates the inaccuracy caused by packet losses. We also derive two specific video quality metrics from MPSNR. The metrics provide a tool for evaluating video streaming over lossy wireless networks.

The rest of the paper is organized as follows. Section II describes related work on video quality measurements. The motivation for developing our new video quality evaluation methodology, MPSNR, is given in Section III. The proposed MPSNR is discussed in Section IV. In Section V, we present experiments that measure the MOS of video streaming in lossy wireless networks and we develop our objective metrics. We compare MPSNR-based metrics with MOS and evaluate their effectiveness in Section VI. Section VII concludes the paper.

## II. Background and Related Work

For most applications, video quality is a subjective term. It is evaluated visually by the viewers. The subjective video quality is measured through each viewer giving a score ranging from one (worst) to five (best). The metric, Mean Opinion Score (MOS), is the arithmetic mean of all these individual scores. However, the measurement of MOS is an expensive process as it needs a large number of viewers and controlled evaluation environments, such as a fixed screen size for displaying a video. It is often impossible to conduct video quality measurements by collecting MOS for every processed video. To cope with this difficulty, objective video quality measurement is used. Objective video quality is based on the criteria and metrics that can be measured objectively and automatically by a computer program. The goal of an objective video quality metric is to approximate the subjective measurement such as the MOS. PSNR is the most widely used objective video quality metric. But due to its inability of approximating the non-linearity of the human visual system, it does not perfectly correlate with the human perceived visual quality. Other complex metrics, such as SSIM [1] and CZD [2] have been proposed to simulate the non-linearity of the human visual system. However, all these metrics were developed from evaluating static image quality, so they are based on a pixel-by-pixel comparison [6]. More importantly, their computation methods do not consider the case in which some frames in the video raw file are lost in the streaming process. Such losses result in mis-alignment of frame sequences in the processed video and the reference video, causing inaccuracies in quality metrics calculation. More complex objective measurements, such as PEVQ [4] and NTIA VQM [5] have been proposed recently. Although they could approximate MOS accurately in general, they still do not explicitly handle frame losses in wireless channels. For example, NTIA VQM requires users to ensure there is no frame missed or dropped in the process; otherwise the quality evaluation will be affected [7].

Besides these traditional and standard metrics, researchers have also proposed other objective metrics. The proposals in [8], [9] are good examples. Engelke *et al* [8] suggested a hybrid image quality metric that extracts different image features, such as blocking, blur, and etc., for video quality evaluation. It is a frame-by-frame video evaluation method. Its simulation results showed a close correlation between the metric and MOS, but the metric does not consider frame loss. Furthermore, these image features extraction algorithms greatly increase the complexity of the video evaluation compared with other frame-by-frame evaluation methods (for example, SSIM [8]). On the other hand, the work of [9] proposed a content-based metric. It evaluates the quality of a video by categorizing the types of content of the video. For each type of content, different parameters are used in the evaluation function. This method avoids the issue of frame loss in the processed video, but it tends to complicate the design and the resource demands in the implementation process.

Due to its simplicity, PSNR still remains the most widely used objective video quality metric. In a recent meeting of International Telecommunication Union (ITU-T), an improved PSNR calculation algorithm was proposed to tackle the problem of constant delay in a processed video [10]. Although it did not tackle the problem of frame losses in the processed video, its approach of finding the corresponding frame can be utilized. We propose our objective video quality evaluation methodology, MPSNR, that enhances the PSNR calculation

[1] The method proposed in this paper is mainly used for evaluating the video streaming capability of wireless networks. Therefore, similar to PSNR, the video evaluation we propose is a full reference (FR) method. That means the reference (original) video is also available in the receiver.

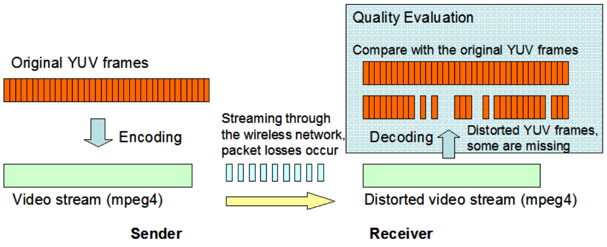

Quality Evaluation
Compare with the original YUV frames
Original YUV frames
Encoding
Streaming through the wireless network, packet losses occur
Decoding
Distorted YUV frames, some are missing
Video stream (mpeg4)
Distorted video stream (mpeg4)
Sender
Receiver

Fig. 1. Illustration of video streaming quality evaluation.

of a video. Using an approach similar to [10], we address the problem of frame losses in the processed video, while retaining the simplicity of the computation. We also use linear regression against MOS to derive two specific metrics from MPSNR.

## III. Motivation for Developing a New Video Quality Evaluation Methodology

### A. *Inaccuracy in the Existing PSNR Calculation*

PSNR, as a video quality measurement, does not accurately indicate the subjective quality of a video. In addition to the effect of the non-linearity of human visual system, the calculation method for PSNR of a video introduces errors in evaluating quality, especially when a video is streamed over a lossy wireless channel.

Traditionally, PSNR is calculated by comparing the first frame of the streamed video (i.e. processed video) with the first frame of the reference video, and then comparing the second frames of the streamed and the reference videos, and so on. This simple calculation method assumes no frames are lost in the streamed video. It works well for evaluating video encoding/decoding errors and video streaming in wired networks, where the frame losses in a video stream rarely occur. However, frame losses are prevalent in wireless networks. In wireless networks, contiguous packet losses could cause the loss of an entire frame in the video stream. (A *frame* in a video is composed of several *packets* in the network layer). Figure 1 shows how a video in the original YUV format is encoded, streamed and converted back to the YUV format for evaluation. Due to packet losses during streaming, some YUV frames are missing after converted from the stream file (for example, an mpeg4 file). A missing frame results in the latter frames in shifted positions when compared with the reference video. The shifted frame position causes incorrect frames to be compared in the PSNR calculations. A human cannot usually detect the loss of a few frames, but the off-position comparisons severely underestimate the average PSNR value of the streamed video.

Figure 2 shows the snapshots of three videos. Figure 2(a) is a snapshot of the reference video that has the “highest” quality. The other two snapshots are from the videos as they are being streamed over a wireless network. The average PSNR value of the reference video is 100dB, that is the highest value. It refers to the case when there are no distortions in any frame of the video. Note that if there is no distortion, the PSNR value should be infinity according to the definition. But for the sake of calculation and analysis, we use the same approach in [8] to define the highest value of PSNR to be 100dB. The average PSNR of video streaming A (Figure 2(b)) is about 38dB, while that of video streaming B (Figure 2(c)) is about 40dB. However, we can clearly see that the quality of the video stream A is much better than that of the video stream B. This example demonstrates how the off-positioned frames (due to the loss of few frames in video A) causes the PSNR to be severely underestimated. This simple example provides the motivation to develop a more comprehensive evaluation method for video streaming in wireless networks. However, it is important to preserve the simplicity of PSNR in any new metric as the expensive hardware and software for complex video evaluation are not always available, and the speed for video evaluation is important especially when there are a large number of videos to be evaluated.

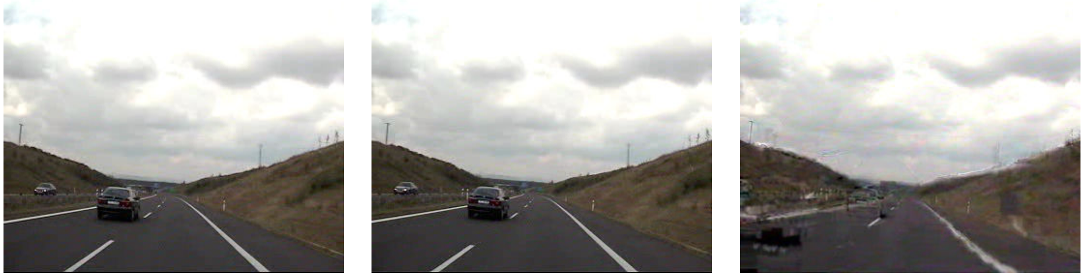
### B. *Brief Description of the New PSNR Calculation*

The error introduced to PSNR calculation due to frame losses in video streaming cannot be easily corrected, as there is no timing information recorded in the raw video frames. Thus, the correct corresponding frame pair from the streaming video and the reference video is not easily identifiable. To solve this problem without introducing significant overhead, we calculate the PSNR of the video frames using a different approach. Instead of ignoring the lost frames and blindly comparing frames from the reference video with those from a streamed video in the order of received, we introduce a “matching” process before determining the “actual” PSNR of the frames in the streamed video. The matching process is critical in our proposed MPSNR (modified PSNR calculation), as it helps us locate the correct frame to compare and calculate the “actual” PSNR value. In Section IV, we discuss an optimized algorithm for matching process. To reduce the complexity of the matching process, and thus the complexity of MPSNR, we also present a heuristic.

In the video stream A (Figure 2(b)), the matching process indicates that 0.3% of the frames are lost, but none of the received frames have any distortion. That means that all the received frames should have the PSNR value of 100dB when compared with the correct corresponding frames in the reference video. Due to the frame losses, the traditional PSNR calculation compares the incorrect frames and returns low PSNR values. However, the proposed MPSNR calculation uses the correct corresponding frames for comparison and returns 100dB of PSNR for every received frame. Therefore, the average PSNR of video streaming A is 100dB. For the video stream B, although there are no frame losses, the received frames have distortion. MPSNR also returns the average PSNR value of about 40dB for video streaming B. This example shows the importance of our matching process in the correct PSNR calculation.

## IV. Anatomy of MPSNR

As an objective video quality evaluation methodology, MPSNR measures PSNR of the streamed (i.e. processed) video



(a) Reference video

(b) Video streaming A

(c) Video streaming B

Fig. 2. Snapshots of different videos.

frames and other network parameters such as loss rate of the video frame, proportion of distorted video frames in a video streaming, etc. These measurements are plugged into linear models, which will be detailed in Section V, to predict the MOS of the video. As mentioned in Section III-B, the matching of correct corresponding frames in the streamed and the reference videos is critical in MPSNR. We first discuss an optimized algorithm for the matching process.

### *A. An Optimized Algorithm for Matching Corresponding Frames*

As shown in Section III, the limitation of the traditional video PSNR calculation is its erroneous pair-up of the corresponding frames from the streamed video and the reference video. An intuitive way to fix this limitation is to incorporate timing information into the raw video, for example the YUV file. However, this approach involves modifications in the decoding mechanism that coverts the streaming file (e.g. mpeg4) to the raw video (e.g. YUV). A number different video coding standards use different coding/decoding mechanisms [11]. Inserting timing information to the raw video is also different from one coding standard to another, and hence increases the complexity in video decoding and affects many other aspects of video processing.

Instead of modifying the decoding mechanism and the raw video file format, we improve the PSNR measurement by introducing a "frame matching process." The matching process helps us locate the correct frame to compare with. We use the similarity of the streamed video and the reference video to find the correct match. First, we make the following assumption.

*The sum of PSNR of all frames in a streamed video is the maximum when all the frames are correctly matched with the corresponding frames in the reference video.* We make this assumption because the corresponding pair of frames should have the greatest similarity and their PSNR value should be the largest among the PSNR values of other unmatched frame pairs. The same assumption was also made in [10] to determine the most probable corresponding frame in the reference video. In [10], the corresponding frame in the reference video is located only for the first frame in the processed (e.g. streamed) video that may experience a constant delay. In our approach, we use this assumption to locate the corresponding frames for all frames in the streamed video. Each frame in a streamed video must have a matched frame in the reference video, and we consider a global maximization of the sum of PSNR. Therefore, the problem of the matching process is stated as:

*Match each frame in a streamed video to a frame in the reference video so that the sum of PSNR of all frame pairs are maximized.*

It is very similar to a *sequence alignment* problem in bioinformatics [12]. In bioinformatics, DNA or RNA sequences are aligned to identify the region of similarity. In our video quality evaluation, the streamed video and the reference video frame sequences are aligned to find the match. The difference is that in sequence alignment, unmatches (called gaps) are allowed in both sequences, while in our case, every frame in the streamed video must find a match in the reference video. Although there is a standard optimized algorithm to solve the sequence alignment problem [13], due to this difference, we need a new algorithm for our use.

We define $OPT(i,j)$ to be the maximum total PSNR value achieved when a streamed video with $j$ frames is matched to the reference video with $i$ frames. Let $psnr(x,y)$ be the PSNR value of frame $x$ and frame $y$. If no match can be found for a frame in the reference video, we ignore the frame in the calculation of the total PSNR value. Figure 3 shows the three possible cases for the last match in two videos. An underline segment indicates no frame is matched. Different from an ordinary sequence alignment, the Case 3 in Figure 3 would never happen as the reference video is always longer than the streamed video. In other words, all frames in the streamed video must find the match in the reference video. But the reverse, that all frames in the reference video must find the match in the streamed video, is not true, so Case 2 is possible. Therefore, the recurrence equation in MPSNR is

$$\begin{aligned} OPT(i,j) = \max[\, & psnr(i,j) + OPT(i-1,j-1), \\ & OPT(i-1,j)\,] \end{aligned} \tag{1}$$

Equation (1) states that when Case 1 is selected, the largest possible total PSNR value for the video is $psnr(i,j) + OPT(i-1,j-1)$. The largest possible total PSNR is $OPT(i-1,j)$ when Case 2 is selected. So, for $OPT(i,j)$, we have to choose the largest among these two possible cases. The recurrence equation (1) shows that similar

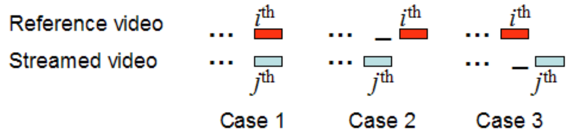


Fig. 3. Three possible cases for the last match.

to a sequence alignment problem, our matching process can be solved by a dynamic programming algorithm [13]. By using dynamic programming, we can find the optimum match of the frames in the streamed video to the frames in the reference video with the maximum total PSNR.

In MPSNR, a frame in the streamed video does not have to compare with every frame in the reference video to find the optimized match. Suppose there are a total of $g$ frames lost during streaming. A frame in the streamed video should only compare with at most $g$ frames in the reference video. To see this, consider frame $q$ in the streamed video. Frame $q$ can only match with a frame between $p+1$ and $p+g$, where frame $p$ matched with frame $q-1$ of the streamed video in the previous iteration of dynamic programming. If frame $q$ matches with any frame beyond frame $p+g$, that implies there are more than $g$ frames lost in the streamed video. It is contradictory to the fact that the total number of frame losses in streaming is $g$. Adding this constraint to the dynamic programming and together with the recurrence equation (1), the time complexity of the optimized matching process in MPSNR is $O(gn)$, where $n$ is the number of frames in the streamed video and $g$ is the total number of frames lost.

### B. A Heuristic Algorithm for Matching Corresponding Frames

Although the time complexity of the optimized matching algorithm is polynomial, the running time can be significant when both the number of frames in the streamed video ($n$) and the number of frame losses ($g$) are large. If a poor wireless channel quality results in a constant loss rate of streaming video and when the length of streaming video increases, the execution time of the matching process will be increased in a much faster rate than the video length because the total number of lost frames also increases. In practice, given a streamed video of 40 seconds (1000 frames) with 20 frames lost (about 2% frame loss rate), a personal computer with 2.8GHz CPU and 1GB RAM needs about 20 seconds to run MPSNR and return the PSNR values of all the frames in the streamed video. The traditional PSNR calculation on the other hand takes less than two seconds for the same video in the same computer. Therefore, we need a faster algorithm for the matching process in MPSNR.

Instead of considering the global maximization of total PSNR in the optimized algorithm, we consider a local maximum PSNR search. Let $\mathit{inPSNR}_{ji}$ be the PSNR value calculated for frame $j$ in the streamed video when it is compared with frame $i$ in the reference video. Frames $i$ and $j$ are not necessary the last frames in the reference video and the streamed video respectively. We use *window* to denote a group of continuous frames in the reference video for the matching process. Let $W_j$ be the set containing the continuous frames in the reference video when frame $j$ in the streamed video is processed. *Window size*, $w$, is the number of continuous frames in $W_j$. Let $\mathit{PSNR}_j$ be the PSNR value of the frame $j$ in the streamed video. $\mathit{PSNR}_j$ is determined using the following.

$$\mathit{PSNR}_j = \max_{i \in W_j}(\mathit{inPSNR}_{ji}) \tag{2}$$

When $\mathit{PSNR}_j$ is determined, we know the frame, say $k$, in the reference video is matched with frame $j$ in the streamed video. At this moment, the window moves. Now, $W_{j+1}$ contains frames from $(k+1)$ to $(k+w)$. The matching process is then carried out for frame $j+1$ in the streamed video. The matching of frame $j$ implies that all the frames that precede frame $k$ in $W_i$ in the referenced video cannot be found in the streamed video (i.e., they are lost in the streaming process). When we perform the matching, we must make sure that the number of remaining frames in the referenced video is no less then the number of remaining frames (the frames that have not gone through the matching process) in the streamed video. Otherwise, some frames in the streamed video cannot match to any reference frame.

According to Equation (2), we take the maximum value of $\mathit{inPSNR}_{ji}$, as the final PSNR, $\mathit{PSNR}_j$, of frame $j$ in the streamed video. It could happen that frame $j$ in the streamed video is distorted severely and has a larger similarity to a non-corresponding frame, $k$, than to the actual corresponding frame, $h$. For example, $\mathit{inPSNR}_{jk} = 10.25$ while $\mathit{inPSNR}_{jh} = 10.19$. In this case, the matching process returns an incorrect corresponding frame. To mitigate this problem, we introduce a parameter called PSNR threshold, *thresh*, into the matching process. We take the maximum $\mathit{inPSNR}_{ji}$, as the final PSNR, only if it is greater than *thresh*. This ensures the returned matched frame has a certain large degree of similarity with the frame $j$ in the streamed video. The larger the PSNR threshold, the more accurate the frame matched. However, if *thresh* is too large, the probability of returning a matched frame from the matching process becomes very small. Even for the corresponding frame pair, the frame in the streamed video could have a certain degree of distortion which decreases the PSNR to be less than *thresh*. If this case happens (i.e., the maximum $\mathit{inPSNR}_{ji}$ is not larger than *thresh*), we will regard the first frame in $W_j$ as the matched frame.

Setting an appropriate *thresh* is not straightforward as it depends on how much the streamed video is distorted and it is unknown before the evaluation. We try different *thresh* values around 30dB for each run of MPSNR and take the largest overall averaged PSNR as the final PSNR value of the streamed video. The reason for choosing 30dB as the mid value is that the distorted frames have an average PSNR of 30dB in lossless streaming. If the maximum $\mathit{inPSNR}_{ji}$ is not less than 30dB, we are confident that frame $j$ is the same frame as frame $i$, but with distortion. In our multihop wireless video streaming environment that will be discussed in Section V-A, we use three different $thresh$ values of 20dB, 30dB and 40dB.

Another important parameter that affects the performance of this algorithm is window size, $w$. If the window size is too small, the "real" matching frame may be outside of the window, and it results in an incorrect match. A large window size has a high probability of finding the correct match, but at the cost of a long computation time. The selection of the window size should consider how much loss the streaming suffers from. In our multihop 802.11 wireless video streaming environment (see Section V-A), a window size of five is large enough.

Using this heuristic matching algorithm we can reduce the time complexity of MPSNR. The time complexity for this heuristic algorithm is $O(twn)$, where $t$ is the number of different *thresh* tried, $w$ is the window size and $n$ is the total number of frames in the streamed video. Both $t$ and $w$ depend on how lossy the wireless channel is. They are constants in a particular wireless system, for example, in our 4-hop wireless network, $t = 3$ and $w = 5$. Although the values of $t$ and $w$ vary from networks to networks, in any given wireless network, $t$ and $w$ are small constants. Therefore, the time complexity of this heuristic matching algorithm is $O(n)$, the same as that of traditional PSNR calculation. Using the same example scenario of a streamed video of 40 seconds (1000 frames) with 20 frames lost (2% frame loss rate), a personal computer with 2.8GHz CPU and 1GB RAM needs about four seconds to run this heuristic in MPSNR and return the PSNR values of all the frames in the streamed video. To further evaluate the effectiveness of this heuristic matching algorithm, we use this heuristic in MPSNR to evaluate video quality and derive quality metrics in Section V. In Section VI, we also derive the metrics from MPSNR using the optimized matching algorithm. The two sets of metrics have similar performances.

### C. Measuring Other Parameters

Calculating PSNR of the video frames is the major function of MPSNR. Along with the PSNR calculation, MPSNR measures other following video streaming related parameters.

- Distorted frame rate ($d$): the percentage of distorted frames (in which the PSNR is less than 100dB) in a streaming video;
- Averaged PSNR of distorted frames (*dPSNR*): the mean PSNR value of all the distorted frames;
- Frame loss rate ($l$): the percentage of lost frames in a streaming video. We derive it from comparing the total number of frames in the received streamed video with that in the reference video;

Once the corresponding frames in a streamed video and the reference video are matched and the PSNR of each frame in the streamed video is calculated, all the above parameters are readily available. In Section V, we use these parameters together with the average PSNR of a video calculated from MPSNR, *aPSNR*, to derive objective metrics for predicting Mean Opinion Score (MOS) of videos.

## V. Developing Metrics from Experiments

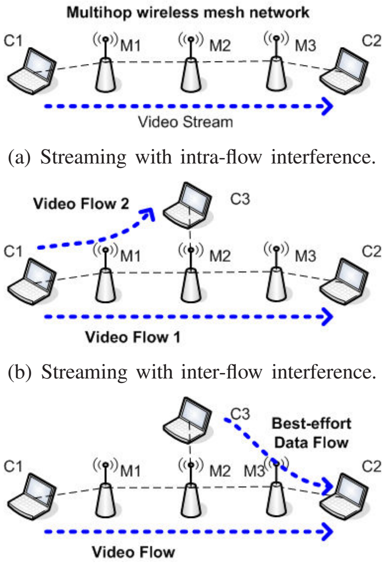
### A. Experiments

*1) Collecting videos of different quality:* We first collect videos from a series of streaming experiments over multihop wireless mesh network [14]. Figure 4 shows the different scenarios in which the video streaming is performed. M1, M2 and M3 are three mesh access points (MAPs). They are mesh routers that relay the network traffic from a client (for example, C1) to another (C2). In our case, C1 is a video streaming server and the video is streamed from C1 to C2 (video streaming client). Figure 4(a) shows a 4-hop wireless mesh network. To collect videos with varying qualities, we configured 3-hop and 2-hop networks as well by removing one and two MAPs, respectively. The degree of intra-flow interference affects the video quality, with longer-hop paths suffering from more interference [15]. We also add inter-flow interference by having another client (C3) receive video streaming from C1 at the same time (Figure 4(b)). In another setting, we add background TCP and UDP data traffic to interfere the video streaming (Figure 4(c)). In each scenario, we also vary the limit of link-layer retransmissions in video streaming. The standard "highway" video [16] is used for streaming because it has constant moving scenes that are sensitive to the frame distortion and loss. Our MPSNR can also apply to videos of other contents. For demonstrating the principle of deriving new quality evaluation metrics from MPSNR, we only focus on the "highway" video in this paper.

Through these experiments, a total of 40 streamed videos with different qualities are collected. We randomly divide these 40 video clips into two groups, a training set and a validation set. We have 30 video clips in the training set that is used to derive the objective video quality metrics. The other 10 video clips form the validation set and they are used to evaluate the effectiveness of the derived objective metrics. It is worth noting that in [9] 39 videos are used for deriving video quality metrics by a linear regression method. Their training set and validation set contain the same set of videos, but with different human subjects to evaluate. We believe this approach is inadequate as different videos could have very different qualities but the scores from different human subjects actually have good agreement. Therefore, different videos in the training set and the validation set give higher confidence in evaluating the performance of the metrics.

*2) Collecting subjective evaluation for video quality:* We engaged 21 volunteers as the subjects to score the quality of every video clip (according to ITU-R BT.500-11 subjective assessment standard [17], at least 15 subjects are needed for subjective quality evaluation, so 21 subjects in our case should be enough). Each subject was asked to score the watched video on a standard five-grade scale [17]. Score 1 is for a video with the worst quality and it means the impairment in the video is very obvious and very annoying. Score 5 is for a video with the best quality and it means the impairment is imperceptible and the video is perfect.

Our test was performed according to the single-stimulus (SS) method [17]. The standard videos with the five different scores were shown to the viewer at the beginning of the test. During the test, only the videos to be scored were shown without any display of the standard/perfect video. For each video clip, we average the quality scores given by the subjects



(a) Streaming with intra-flow interference.

(b) Streaming with inter-flow interference.

(c) Streaming with background data flow.

Fig. 4. Video streaming in different scenarios.

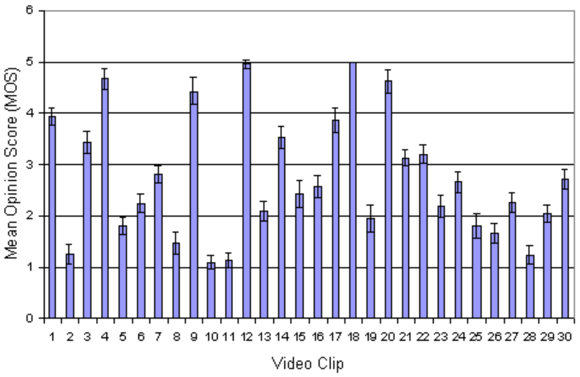


Fig. 5. MOS and 95% confidence intervals of videos in the training set.

and obtain a mean score that is the Mean Opinion Score (MOS).

Diversity was taken into account when we chose the test volunteers. The age of our volunteers ranges from 20 to 45. Eight of them (38.1%) are female while 13 are male. Their occupation ranges from university undergraduate students to laboratory technicians. Each subject was asked to score the same set of video clips (but in different sequence order) twice. To avoid unreliable and inconsistent results, for each video, if scores from a particular subject in the two rounds of experiments differed by two or more, the score from this specific subject is discarded. Throughout the entire test, 1.19% of the scores were rejected under this condition.

Figure 5 shows that the MOS of the videos in the training set ranges from 1.095 to 5. This shows that we have chosen a set of videos with a wide range of quality. The averaged size of 95% confidence interval among the videos in the training set is 0.38 in the 1 to 5 MOS scale. This indicates a good agreement among the subjects.

### *B. Deriving Metrics from Subjective Evaluation and MPSNR*

*1) POMOS:* By applying MPSNR with the heuristic matching algorithm to the videos in the training set, we first obtain the *aPSNR* (the average PSNR calculated from MPSNR) and the traditional PSNR (*tPSNR*) for each video. Noted that traditional PSNR of the video can also be obtained from MPSNR by setting the window size ($w$) to one. Figure 6(a) shows the scattered-plot of MOS for both *aPSMR* and *tPSNR* of each video in the training set. Compared with *tPSNR*, *aPSNR* demonstrates a more consistent relationship with MOS. Although the *tPSNR* also demonstrates a linear trend with MOS when PSNR values are small, they deviate significantly when the PSNR gets larger. Thus, the mapping of traditional PSNR to MOS does not hold. However, *aPSNR* demonstrates a close-to-linear relationship with MOS. Hence, we use linear regression to predict MOS of a video from its *aPSNR*.

We propose a two-parameter linear model to predict MOS.

$$\mathit{POMOS} = \beta_0 + \beta_1 \mathit{aPSNR} \tag{3}$$

for some constants $\beta_0$ and $\beta_1$. In this linear model, we use the average PSNR, *aPSNR*, calculated from MPSNR as the predictor variable. *POMOS* is the predicted MOS, not the actual MOS that is evaluated from the human subjects. Hence, *POMOS* is an objective video quality metric ("objective MOS") based on *aPSNR*.

Since *POMOS* itself is already a mean value (as MOS is a mean value), the error term, $\epsilon$, that is usually added in a regression analysis can be dropped [18]. If we predict a quality score, $Y$, given by a particular user, we have

$$Y = \beta_0 + \beta_1 X + \epsilon \qquad E[\epsilon] = 0 \tag{4}$$

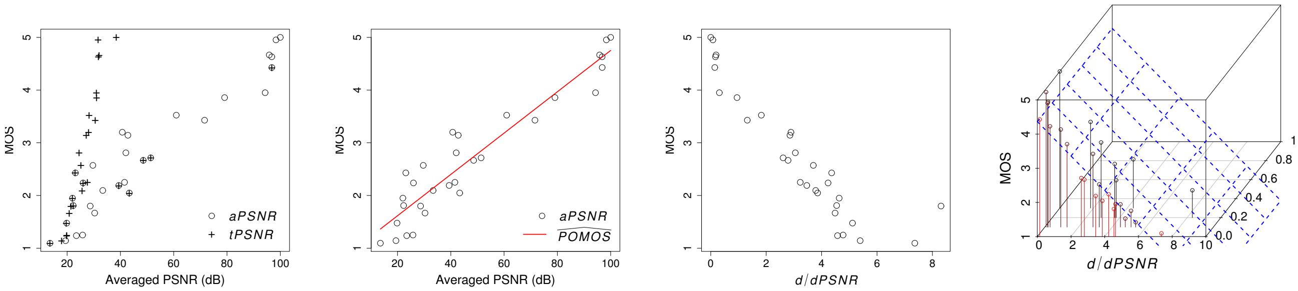
where $X$ can be any predictor variable. We are only interested in predicting the mean value of $Y$ that is *POMOS*, hence we ignore the error term, $\epsilon$.

Figure 6(b) shows the linear fit of the estimated *POMOS*, $\widehat{\mathit{POMOS}}$. We use the *linear model* package of the statistics tool, ***R*** [19], to derive $\hat{\beta}_0$ and $\hat{\beta}_1$, that are respectively the estimates of $\beta_0$ and $\beta_1$ in Equation (3). The final linear equation for estimating MOS is

$$\widehat{\mathit{POMOS}} = 0.8311 + 0.0392 \mathit{aPSNR} \tag{5}$$

The 95% confidence interval for $\hat{\beta}_1$ is (0.03431, 0.04411). The small interval indicates that the sample size (number of videos) in the training set is large enough for a good estimation. Mean of $\hat{\beta}_1$ (0.0392) is significant to *POMOS* prediction as *POMOS* ranges only from 1 to 5 while *aPSNR* ranges from 0 to 100. This justifies our decision of including *aPSNR* in our linear model for predicting MOS.

*aPSNR* is a mean value over all PSNR values of the frames in a video clip calculated from MPSNR. According to the definition of PSNR, if the received frame has no distortion compared with the corresponding frame in the referenced video, the PSNR value of this perfect frame is infinity. For calculation of *aPSNR*, we must give a finite value for the



(a) Scatter plot of MOS for $aPSNR$ and $tPSNR$.

(b) Linear fit of POMOS.

(c) Scatter plot of MOS against $d/dPSNR$.

(d) Linear plane fit of ROMOS.

Fig. 6. Mapping of different measurements of videos to MOS.

PSNR of such frame. Therefore, we assigned a PSNR of 100dB for the perfect frames. However, the PSNR value of such perfect frames affects *aPSNR* and in turn the MOS prediction. To mitigate this problem, we develop another linear model that does not use the PSNR value of the perfect frames.

*2) ROMOS:* As in Section IV-C, we define *dPSNR* as the averaged PSNR of all the distorted frames in a streamed video, and $d$ as the distorted frame rate. The video quality decreases as *dPSNR* decreases, but the video quality also decreases as $d$ increases. From Figure 6(c), we find that as the ratio of distorted frame rate to averaged PSNR of distorted frames ($d/dPSNR$) increases, MOS of the video decreases. Therefore, instead of using *aPSNR*, we use $d/dPSNR$ in our linear model to predict MOS of a video. For those lost frames, they are neither perfect frames nor distorted frames. We must take the lost frames into account in the prediction of MOS. Thus, we include the frame loss rate, $l$, in the prediction. Finally, we have our linear model of MOS prediction as Equation (6).

$$ROMOS = \beta_0 + \beta_1 \frac{d}{dPSNR} + \beta_2 l \qquad (6)$$

Like *POMOS*, *ROMOS* is an objective video quality metric, but it is based on rates $d$ and $l$. Figure 6(d) shows the plane fits the scatter MOS values. Again, we use the *linear model* package of ***R*** to derive $\hat{\beta}_0$, $\hat{\beta}_1$ and $\hat{\beta}_2$, that are respectively the estimates of $\beta_0$, $\beta_1$ and $\beta_2$ in Equation (6). The final linear equation for estimating MOS is

$$\widehat{ROMOS} = 4.367 - 0.5040 \frac{d}{dPSNR} - 0.0517l \qquad (7)$$

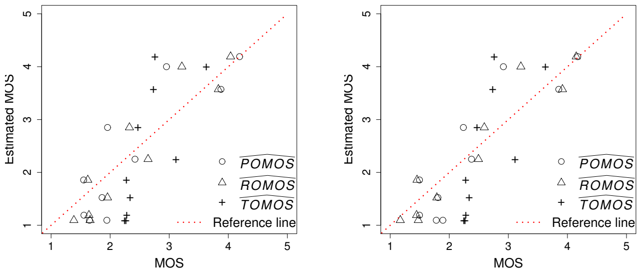
where $\widehat{ROMOS}$ is the estimated *ROMOS* from our linear model (6). The 95% confidence interval for $\hat{\beta}_1$ is (-0.58902, -0.41894). The small interval indicates that the sample size (the number of videos) in the training set is large enough for a good estimation. Mean of $\hat{\beta}_1$ (-0.5040) is significant to *ROMOS* prediction as *ROMOS* ranges only from 1 to 5 while $d/dPSNR$ in our case ranges from 0 to 8. This justifies the inclusion of $d/dPSNR$ in our linear model. For $\hat{\beta}_2$, its mean is -0.0517 and the 95% confidence interval is (-0.15428, 0.05098). Its mean is close to zero and its 95% confidence interval is large. These imply that the inclusion of $l$ is not significant for the prediction of MOS and the sample size in training set is not large enough to show the significance of $l$ in prediction. The reason is that the frame loss rate is often small (around 0.2%) in our wireless video streaming experiments. Such a small frame loss rate causes significant inaccuracy in traditional PSNR calculation, but it does not greatly affect the subjective quality evaluation. However, in some other wireless scenarios, the frame loss rate may be much severe, and hence we include it in our linear model for predicting MOS.

## VI. Evaluation of Objective Metrics

In Section V-B, we use the 95% confidence interval of the estimated coefficients of the linear models to evaluate the effectiveness of different predictor variables. In this section, with the help of the validation set of videos, we evaluate the accuracy of our newly developed objective video quality metrics. We first find the MOS of each video in the validation set by recording all the quality scores rated by the 21 subjects. For each video in the validation set, we then calculate $\widehat{POMOS}$ and $\widehat{ROMOS}$ from Equations (5) and (7) respectively. For comparison, we also develop a linear model for predicting MOS from the traditional PSNR (*tPSNR*).

$$TOMOS = \beta_0 + \beta_1 tPSNR \qquad (8)$$

The model is similar to Equation (3), with traditional PSNR *tPSNR* replacing *aPSNR* calculated from MPSNR. We find the Pearson correlation (also known as correlation coefficient) [3] between the MOS and the estimated MOS values from $\widehat{TOMOS}$, $\widehat{POMOS}$ and $\widehat{ROMOS}$, where $\widehat{TOMOS}$ is an estimated value of Equation (8). Pearson correlation is used to evaluate the prediction accuracy of the linear models. The higher the correlation, the more accurate the prediction. For $\widehat{POMOS}$, it has a Pearson correlation of 0.8666 with MOS. For $\widehat{ROMOS}$, it has an even higher Pearson correlation of 0.9346. Although we see a linear trend of MOS against traditional PSNR in Figure 6(a), the Pearson correlation of $\widehat{TOMOS}$ with MOS is only 0.7274. Figure 7(a) visualizes the relationship between the actual MOS from 21 subjects and the estimated MOS values from the objective calculation of different linear models.



(a) From heuristic matching.

(b) From optimized matching.

Fig. 7. Scatter plot of estimated (objective) MOS values against actual (subjective) MOS.

We can see that for the videos in the validation set, $\widehat{ROMOS}$ are the closest to the reference line and $\widehat{TOMOS}$ are the farthest from the reference line. Compared with [8] that also used the "highway" video clip as the evaluation video, their objective video quality evaluation metric only achieves a Pearson correlation of 0.896. The content based metric in [9] although classify the content categories of the video, its averaged Pearson correlation is only 0.8303. Furthermore, their objective metrics have much higher complexity than ours as we use a simple pixel-by-pixel PSNR calculation algorithm.

We now change the matching algorithm in MPSNR from heuristic to optimized, and again perform the derivation of metrics. As expected, the Pearson correlation of the metrics with the MOS increases, but the improvement is not significant. For $\widehat{POMOS}$, it has a Pearson correlation of 0.8838 with MOS while for $\widehat{ROMOS}$, 0.9509. The scatter plot of these estimated MOS values against MOS is shown in Figure 7(b), that is quite similar to Figure 7(a). This similarity shows that our heuristic matching algorithm works very well.

It is worth noting that the coefficient values we derived in Equation (5) for $\widehat{POMOS}$, and in Equation (7) for $\widehat{ROMOS}$ are specific for videos with the content belonging to the same category as "highway" video. According to [9], there are only five different video content categories and technologies exist to classify the content category of a video. By following the same procedure in Section V to derive the coefficient values of Equation (3) and Equation (6) for each content category, we can apply *POMOS* and *ROMOS* to all other videos.

## VII. Conclusion

Traditional PSNR calculation overlooks the packet loss in wireless networks, and hence it is not an adequate method to compute PSNR of video streaming over wireless networks. We develop a novel video quality evaluation methodology, MPSNR, to address the shortcomings of the traditional method. By matching the correct frame pairs in the streamed video and the reference video, MPSNR calculates accurate PSNR of the streamed videos. From human subjective video evaluations, we find that the PSNR value calculated from MPSNR demonstrates a close-to-linear relationship with the subjective MOS. Using linear regression, we derive an objective video quality metric, *POMOS*, based on PSNR value to predict the MOS of a video. *POMOS* has a high Pearson correlation of 0.8664 with the MOS. Adding other video streaming measurements, such as the proportion of distorted frames in video streamings, we derive a more comprehensive metric, *ROMOS*, to predict the MOS of a video. *ROMOS* has a Pearson correlation of 0.9350 with the MOS. Both metrics assess the video quality more accurately than the traditional PSNR while retaining the simplicity of PSNR. With the popularity of video applications in wireless networks, these two metrics provide a significant tool for evaluating the performance of such applications. Based on the correct video quality evaluation, we expect further advancement of video over wireless network technologies.

## References

[1] Z. Wang, L. Lu, and A. Bovik, "Video quality assessemnt based on structural distortion measurement," *Signal Processing: Image Communication*, vol. 19, no. 2, pp. 121–132, February 2004.

[2] I. Avcibas, N. Memon, and B. Sankur, "Steganalysis using image quality metrics," *IEEE Transactions on Image Processing*, vol. 12, no. 2, pp. 121–129, February 2003.

[3] J. L. Rodgers and W. A. Nicewander, "Thirteen ways to look at the correlation coefficient," *American Statistician*, vol. 42, pp. 59–66, 1988.

[4] OPTICOM, *PEVQ Advanced Perceptual Evaluation of Video Quality*. OPTICOM GmbH, Germany: PEVQ Whitepaper, 2005.

[5] M. H. Pinson and S. Wolf, "A new standardized method for objectively measuring video quality," *IEEE Transactions on Broadcasting*, vol. 50, no. 3, September 2004.

[6] A. Rossholm and B. Lovstrom, "A new low complex reference free video quality predictor," in *IEEE Workshop on Multimedia Signal Processing*. IEEE, 2008, pp. 769–772.

[7] M. H. Pinson and S. Wolf, *Batch Video Quality Metric (BVQM)*. Department of Commerce, USA: NTIA Handbook HB-09-441c, 2009.

[8] U. Engelke, T. M. Kusuma, and H.-J. Zepernick, "Perceptual quality assessment of wireless video applications," in *ITG Source and Channel Coding*. VDE Verlag GmbH, 2006.

[9] M. Ries, O. Nemethova, and M. Rupp, "Video quality estimation for mobile h.264/avc video streaming," *Journal of Communications*, vol. 3, no. 1, pp. 41–50, January 2008.

[10] S. Wolf and M. H. Pinson, "Reference algorithm for computing peak signal to noise ratio (psnr) of a video sequence with a constant delay," in *ITU-T Contribution COM9-C6-E*. ITU, February 2009.

[11] M. Ghanbari, *Video coding: an introduction to standard codecs*. Michael Faraday House, Six Hills Way, Stevenage, Herts. SG1 2AY, United Kingdom: The Institution of Electrical Engineers, 1999.

[12] D. M. Mount, *Bioinformatics: Sequence and Genome Analysis (2nd edition)*. Cold Spring Harbor, New York, USA: Cold Spring Harbor Laboratory Press, 2004.

[13] J. Kleinberg and E. Tardos, *Algorithm Design*. USA: Addison-Wesley, 2005.

[14] A. Chan, S.-J. Lee, X. Cheng, S. Banerjee, and P. Mohapatra, "The impact of link-layer retransmissions on video streaming in wireless mesh networks," in *Proceedings of Internation Wireless Internet Conference*. ACM, 2008.

[15] X. Cheng, P. Mohapatra, S.-J. Lee, and S. Banerjee, "Performance evaluation of video streaming in multihop wireless mesh networks," in *NOSSDAV '08: Proceedings of the 18th International Workshop on Network and Operating Systems Support for Digital Audio and Video*. New York, NY, USA: ACM, 2008, pp. 57–62.

[16] Video, "Arizona state university, video traces research group, qcif sequences," http://trace.eas.asu.edu/yuv/qcif.html, 2009.

[17] ITU, *Methodology for the subjective assessment of the quality of television pictures*. International Telecommunication Union: Recommendation ITU-R BT.500-11, 2002.

[18] N. Matloff, *From Algorithm to Z-Scores: Probabilistic and Statistical Modeling in Computer Science*. http://heather.cs.ucdavis.edu/matloff/probstatbook.html: Creative Commons Lincense, 2009.

[19] R, "The r project for statistical computer," http://www.r-project.org, 2009.