October 2004

Application-Specific Path Switching: A Case Study for Streaming Video

Shu Tao
University of Pennsylvania

Roch A. Guérin
University of Pennsylvania, guerin@acm.org

Follow this and additional works at: http://repository.upenn.edu/ese_papers

Recommended Citation
Application-Specific Path Switching: A Case Study for Streaming Video

Abstract
The focus of this paper is on improving the quality of streaming video transmitted over the Internet. The approach we investigate assumes the availability of multiple paths between the source and the destination, and dynamically selects the best one. Although this is not a new concept, our contribution is in estimating the "goodness" of a path from the perspective of the video stream, instead of relying only on raw network performance measures. The paper starts by showing that the use of raw network performance data to control path switching decisions can often result in poor choices from an application perspective, and then proceeds to develop a practical approach for evaluating, in real-time, the performance of different paths in terms of video quality. Those estimates are used to continuously select the path that yields the best possible transmission conditions for video streaming applications. We demonstrate the feasibility and performance of the scheme through experiments involving different types of videos.

Keywords
path switching, quality, streaming video

Comments
Copyright ACM, 2004. This is the author's version of the work. It is posted here by permission of ACM for your personal use. Not for redistribution. The definitive version was published in Proceedings of the 12th Annual ACM International Conference on Multimedia 2004, pages 136-143. Publisher URL: http://doi.acm.org/10.1145/1027527.1027553

This conference paper is available at ScholarlyCommons: http://repository.upenn.edu/ese_papers/79
Application-Specific Path Switching: A Case Study for Streaming Video

Shu Tao and Roch Guérin
Department of Electrical & Systems Engineering
University of Pennsylvania, Philadelphia, PA 19104
{shutao@seas, guerin@ee}.upenn.edu

ABSTRACT

The focus of this paper is on improving the quality of streaming video transmitted over the Internet. The approach we investigate assumes the availability of multiple paths between the source and the destination, and dynamically selects the best one. Although this is not a new concept, our contribution is in estimating the “goodness” of a path from the perspective of the video stream, instead of relying only on raw network performance measures. The paper starts by showing that the use of raw network performance data to control path switching decisions can often result in poor choices from an application perspective, and then proceeds to develop a practical approach for evaluating, in real-time, the performance of different paths in terms of video quality. Those estimates are used to continuously select the path that yields the best possible transmission conditions for video streaming applications. We demonstrate the feasibility and performance of the scheme through experiments involving different types of videos.

Categories and Subject Descriptors

C.2.1 [Computer-Communication Networks]: Network Architecture and Design; H.4.3 [Information Systems Applications]: Communications Applications

General Terms
Design, Experimentation, Performance

Keywords
Path Switching, Quality, Streaming Video

1. INTRODUCTION

The steady rise in the quality and availability of bandwidth across the Internet has made the transmission of streaming video a reality. However, in spite of this progress, sources of impairment are still present and devising mechanisms capable of further improving the performance of video transmission remains a worthy goal. Such a goal has been the focus of many previous works, which we briefly review here to highlight how they differ from this paper and its contributions.

In general, most earlier works targeted adapting the encoding or the transmission of the video stream in response to changes in network performance. For example, rate control [7, 11] can be used to dynamically adjust the transmission rate of a video flow when path condition fluctuates. Occasional delay and loss variations can be concealed by adapting the rate of retrieving data from the playout buffer and the rate of presenting the decoded media to the viewer [8]. Similarly, upon detecting severe congestion, the receiver can signal the sender to reduce its encoding and transmission rate to maximize the likelihood that the quality of the received video remains acceptable to the viewer [8, 2, 12]. More recently, the use of path diversity has also been studied as an option to provide an extra dimension of adaptability to video applications. For instance, Apostolopoulos et al. [3] investigated an approach that relies on the simultaneous transmission of several substrings of the video signal over different paths, where each substring encodes a partial description of the video. The video can be correctly decoded with graceful quality degradation, even if some of the substrings are missing or incomplete.

Our study shares with these works the fact that it does not require the introduction of additional network-based mechanisms, but involves the participation of the video application (or an access gateway acting on behalf of the application). Meanwhile, our scheme differs from these existing works in that we do not attempt to adapt either the encoding or the transmission of the video data. Instead, we consider an environment that offers path diversity, namely, the ability to select from among multiple possible paths when transmitting video packets between a given source and destination, and investigate how to dynamically select the best path in order to optimize the quality of the received video. In other words, we seek to determine the best conditions under which the network and the application can interact.

Our investigation was motivated in part by the increased availability of multi-homing and overlay networks that have made path diversity more common, and by a number of recent works [1, 16] that have demonstrated that simple path switching strategies can improve the performance of end-to-end data transfers. However, applying the path selection strategies proposed in those works is not necessarily beneficial for improving video quality, because those strategies were developed with the goal of optimizing network performance. For example, the method of [16] makes path switching decisions to achieve the lowest overall end-to-end loss. Therefore, it selects paths based on their predicted loss rates. This does not always result in improved video quality, since video quality is influenced by multiple characteristics of the loss process, not just loss rate. For example, the burstiness of losses, the distance between
from the perspective of video quality, and then use this information to build a path switching mechanism that maximizes video quality. However, such a solution is still inadequate for our purpose, not because of the difficulty in having both the original and the received video frames be simultaneously available at the sender where path switching decisions are being made. We demonstrate those improvements through a number of experiments involving a variety of video streams and different types of network impairments.

The remainder of the paper is organized as follows. Section 2 introduces the distortion model used to relate network performance, namely losses, and video quality. Section 3 briefly reviews a network probing method we previously developed, and on which we rely to infer the losses that the video stream would experience on different paths. Section 4 combines the results of Section 2 and 3 in devising and evaluating a path switching strategy that improves video quality by selecting the best performing path. Finally, Section 5 summarizes the findings of the paper and outlines directions for future work.

2. LOSS-QUALITY MODEL

In this section, we describe the loss-quality model we use to estimate video quality under different loss conditions. The method is derived from the methodology proposed in [14, 9]. An exact model obviously depends on the implementation details of the video transmission system. In our analysis, we use an internally developed MPEG-1/2 video streaming software as the reference implementation. However, as shown in this section, the model can be tuned to fit other implementations as well.

2.1 Basic distortion model

Consider a video sequence with frame size $N_1 \times N_2$, we use $f[k]$ (of size $N_1 \times N_2$) to represent the original video frame, and $\hat{f}[k]$ to denote the corresponding frame restored at the receiver’s side. Thus, the error signal in frame $k$ is

$$e[k] = \hat{f}[k] - f[k],$$

which represents the signal impairment in frame $k$ incurred by transmission loss. A frequently used metric for measuring the distortion is the Mean Square Error (MSE), which is defined as

$$\sigma^2[k] = (e^T[k] \cdot e[k])/(N_1 \cdot N_2).$$

The total distortion of the video is the MSE averaged over all frames. Packet losses are handled as follows: if the receiver detects any number of packet losses in a frame, it discards the entire damaged frame and the most recently decoded frame is repeated. For instance, if any error happens in frame $k$ while frame $(k-1)$ has been correctly received, $\hat{f}[k] = \hat{f}[k-1]$ and $e[k] = f[k-1] - f[k]$.

An important issue in modeling the distortion with motion compensated video coding is the propagation of error signals. Since such video encoding schemes introduce decoding dependencies between adjacent frames, a packet loss affects not only the frame missing the data contained in that packet, but also other frames with decoding dependencies on it. However, because of the spatial filtering effect of the decoder (which can be modeled as a low pass filter [14]), the error signal tends to decay over time. If an error occurs in frame $k$ with an MSE of $\sigma^2[k]$, the power of the propagated error signal is further reduced.
error at frame \((k + i)\) can be approximated as \[\sigma[k + i]^2 = \sigma[k]^2 \cdot \gamma^i.\] (3)

The attenuation factor \(\gamma (\gamma < 1)\) accounts for the effect of spatial filtering, and therefore is dependent on the power spectrum density of the error signal and the impulse response of the loop filter contained in the decoder.

To prevent error propagation, periodic intra-frame coding is used in MPEG-1/2. As a result, error in one frame can only propagate to the frames following the damaged frame in the same group of picture (GoP). If the GoP starts with an I-frame and continues with \((T - 1)\) P-frames\(^1\), the total distortion caused by a single loss is

\[D = \sum_{i=0}^{x-1} \sigma[k + i]^2,\] (4)

where \(x\) is the number of frames from where the original loss occurs (frame \(k\)) to the end of the GoP. We assume that \(x\) is uniformly distributed in \([0, T - 1]\), and that the initial error caused by losing one frame is a constant, i.e., \(\sigma[k]^2 = \sigma_S^2\). Thus, the total distortion caused by losing a single frame is approximately \([9]\)

\[D_2 = \sum_{i=0}^{T-1} \sigma_S^2 \cdot \gamma^i (1 - \frac{i}{T}),\] (5)

Here, \(\alpha\) is a function of \(\gamma\) and \(T\) that accounts for the total propagation effect of the error signal. Although seemingly restrictive, the assumption that the initial error caused by a single loss is constant is reasonable, because we are mainly interested in the average distortion over the whole video sequence instead of the distortion in individual frames. For the same reason, we will also assume that \(\alpha\) is equal to its average value throughout the frame sequence. The exact value of \(\alpha\) can be estimated by simulating individual losses and measuring the MSE in the decoded frames. Nevertheless, as shown in Section 4, estimating the value of \(\alpha\) is not necessary for path selection.

### 2.2 The effect of loss pattern

The basic distortion model only considers the effect of a single loss. In [14], the total loss effect is modeled as a product of the loss rate and the average distortion caused by a single loss. This assumes that the effects of individual losses are independent of each other. However, in practice the video flow may experience various loss patterns and in particular losses can be temporally correlated. Therefore, it is important to understand the impact of different loss patterns on the quality of the transmitted video.

First, packet loss could be bursty, i.e., a single loss event or loss burst may consist of a number of consecutive losses. In our video transmission system, when a loss burst falls in a single frame, the resulting distortion will always be the distortion caused by missing that frame no matter how many packets are contained in the loss burst. When a loss event affects several consecutive frames, we simply model the total distortion as the superposition of the distortion of missing each individual frame. As pointed out in [9], this is an optimistic approximation, as the total distortion in this case is typically greater than the approximated value. For example, the distortion of losing 2 consecutive frames \((k - 1)\) and \(k\) (i.e.,

\(^1\)To simplify the analysis, this model does not consider the bi-directionally predicted frames or B-frames.

\(^2\)The other extra factor \(\sigma_S^2\) turns out to be small compared with the total distortion and therefore can be ignored.
To represent video quality, it can be computed from the value of

\[ T \text{ to further illustrate the effect of loss pattern on video quality, we } \]

that models the total effect of loss on video quality. If we use PSNR

the impaired video to the viewer. We evaluate the combined ef-

fect of loss burstiness and loss distance as follows. Let

average distortion caused by a single loss event:

Here \( r \) describes further that simple loss rate measurements are not suf ficient

than a Bernoulli process with the same loss rate. The figure illus-

a 1% bursty loss rate yields a PSNR level that is about 2.5 dB better

to assess the supportable video quality on a path.

3. LOSS ESTIMATION

Our quality estimation model relies on the online characteriza-

ation of the loss process that the video flow would experience, and in

this section we outline our approach for acquiring this information. In [15], we proposed a solution based on a Hidden Markov Model (HMM) of the evolution of path state. Here, we briefly review the resulting loss model and highlight our methodology for estimating the required loss parameters. Interested readers are referred to [15] for detailed discussions.

We use a 2-state HMM to characterize the loss process experienced by a flow. In this model, the state of a network path seen by the application flow can be either congested (\( S_1 \)) or not (\( S_2 \)), each having a certain loss probability, i.e., \( b_1 \) and \( b_2 \) (\( b_1 > b_2 \)). The transition probabilities between \( S_1 \) and \( S_2 \), \( a_{12} \) and \( a_{21} \), statistically model the evolution of path state. Note that the state evolution of a path is a continuous-time process, while different flows can be considered as sampling this process at different time granularities. Therefore, we can first use a probing flow (with relatively large probing interval to reduce probing overhead) to sample the path, and then a discrete-time HMM representing the loss process experienced by the application flow can be obtained given the sampling intervals of both the probing flow and the application flow.

From the inferred discrete-time HMM, we can compute the loss parameters that are required for video quality estimation. For example, we can first derive the steady state probabilities \( \pi = [\pi_1, \pi_2] \), where \( \pi_i \) represents the probability of the path being in state \( i \) (\( i = 1, 2 \)). Then, loss burstiness can be estimated by computing the probability of seeing a loss event composed of \( l \) consecutive

![Figure 3: The difference in loss factor for the loss models used in Fig. 1.](image1)

![Figure 4: An illustrative example of the impact of loss pattern on video quality.](image2)

<table>
<thead>
<tr>
<th>Loss model</th>
<th>Bursty loss percentage</th>
<th>PSNR difference (dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1% Bernoulli loss</td>
<td>1% Bernoulli loss</td>
<td>4% Bernoulli loss</td>
</tr>
<tr>
<td>2% Bernoulli loss</td>
<td>2% Bernoulli loss</td>
<td>4% Bernoulli loss</td>
</tr>
<tr>
<td>3% Bernoulli loss</td>
<td>3% Bernoulli loss</td>
<td>4% Bernoulli loss</td>
</tr>
<tr>
<td>4% Bernoulli loss</td>
<td>4% Bernoulli loss</td>
<td>4% Bernoulli loss</td>
</tr>
<tr>
<td>5% Bernoulli loss</td>
<td>5% Bernoulli loss</td>
<td>4% Bernoulli loss</td>
</tr>
</tbody>
</table>

\[ PSNR = 10 \log_{10} \frac{255^2}{D} = 10 \log_{10} \frac{255^2}{\xi \cdot L \cdot \alpha \cdot \sigma_S^2}. \]
that the switching process itself can have on video quality. In our environment, e.g., see [16], path switching is conducted by an access gateway that is responsible for measuring the network, selecting the path, and forwarding video packets onto the selected path. Path switching is, therefore, transparent to both the video sender and receiver. Nevertheless, this process may affect video quality if there are differences in end-to-end delay between the candidate paths. Therefore, it is important to understand if and how this can impact video quality.

Assume that a video stream is switched from path A to path B, while its sender keeps transmitting packets at a constant rate of \( r \) packets per unit of time. When the propagation delay on path \( A(d_A) \) is smaller than on path \( B(d_B) \), path switching causes a reception “gap” of duration \((d_B - d_A)\) at the receiver. Conversely, when \( d_A > d_B \), path switching will generate a burst of out-of-order packets at the receiver, whose size depends on both \( r \) and \( (d_A - d_B) \). Reception gaps and out-of-order packets are certainly undesirable, but most current video streaming systems include mechanisms that adapt the occupancy level of their playout buffer in order to accommodate jitter and out-of-order packet delivery. For example, a common playout buffer design [13] involves a combination of two watermarks that define a range outside of which buffer overflow and underflow can occur, and two thresholds that identify a target area in which buffer occupancy should be maintained. A number of effective methods have been developed for maintaining buffer occupancy in the target area across a broad range of network perturbations. For example, the receiver can reduce or increase the frame presentation rate [13], or signal the sender to adjust its rate [5]. Given that most video streaming applications enforce a pre-buffering of a few seconds of playout data [8], while the end-to-end delay difference between network paths is typically much smaller, one can expect the impact of path switching to be easily hidden by those existing mechanisms.

### 4.2 Comparing path quality

As shown in Eqs. 10 and 11, the parameters \( \xi, L, \alpha, \) and \( \sigma_\delta^2 \) allow us to estimate the PSNR of a video signal as if it were sent over a path with known characteristics. As discussed in Sections 2 and 3, \( L \) can be readily obtained and the loss factor \( \xi \) at each available path can be estimated through active probing and statistical inference. However, accurately estimating \( \alpha \) and \( \sigma_\delta^2 \) is non-trivial, as it involves understanding how losses propagate while decoding a video stream and measuring the resulting average distortion [14].

This is typically difficult in live video streaming systems. Fortunately, in the context of path switching, estimating the exact values of \( \alpha \) and \( \sigma_\delta^2 \) is not required, because we are mainly interested in the *quality difference* between two (or more) paths. Therefore, when computing the difference in PSNR between two paths, the only contributing parameters are the loss factors of the two paths, which can both be estimated in real-time. Specifically, if the same video is transmitted on two paths that have loss factors \( \xi_1 \) and \( \xi_2 \) respectively, the quality difference in PSNR is

\[
\text{PSNR}_{\text{diff}} = \text{PSNR}_{10} \frac{\xi_1}{\xi_2}
\]

To verify the accuracy of this expression in estimating differences in quality, we used 30 different loss models spanning a broad range of loss characteristics and various values for the loss factor \( \xi \). Using each loss model, we generated a loss trace and applied it to the video streams *Alberta* and *Robot*. For each pair of loss trace and received video, we computed the value of \( \text{PSNR}_{\text{diff}} \) using Eq. 15 and compared it with the measured PSNR difference. The scatter-plots of this comparison are shown in Fig. 6. Clearly, the quality
The difference estimated by Eq. 15 is close to the actual value for both video sequences. Note that our model tends to underestimate the quality difference when the ratio of $\xi_1/\xi_2$ is large (the upper right region in the plot). This is because the model is built upon the assumption that individual loss events are far apart so that the distortions caused by different loss events do not interfere with each other. This assumption becomes weaker as losses become heavier or $p_e$ becomes larger. However, this inaccuracy typically will not affect the path switching decisions if the threshold of quality difference used for path switching is appropriately selected.

The last step in formulating a practical strategy for real-time path switching that optimizes video quality is to confirm that differences in PSNR are indeed an accurate measure of differences in quality, and can therefore be used to make correct path selections. In particular, although a positive PSNR difference does imply that one path is better, the difference may not be meaningful in terms of quality and could even be within the error margin of our methodology. It is therefore desirable to identify when PSNR differences correspond to significant enough differences in video quality. Furthermore, several studies have shown that PSNR is a good measure of the actual (or subjective) video quality only within a certain range of values, and that the relationship between the two is highly non-linear [17]. For example, a proposed mapping between PSNR and a Mean Opinion Score (MOS) ranging from 0 to 1 (0 represents the best quality, 1 represents the worst quality) is as follows [17]:

$$MOS = \frac{1}{1 + \exp(b_1(PSNR - b_2))},$$  \hspace{1cm} (16)

where in [18], the following values were suggested for $b_1$ and $b_2$: $b_1 = 0.1701$, and $b_2 = 25.6675$.

In order to better assess the relationship between PSNR and video quality, how it evolves for different PSNR values, and how it varies for different video streams, we present in Fig. 7(a) sample PSNR-quality mappings for Alberta and Robot. The results were obtained by comparing the original and impaired frame sequences using the objective video quality metric (VQM) developed by the Institute for Telecommunication Science [18]. From the figure, we clearly see that the relationship between PSNR and quality is highly non-linear, and the mapping trend is roughly the same for both videos. This illustrates that even if a general expression such as Eq. 16 is not perfectly accurate, it does capture the relationship between PSNR and video quality. This allows us to finalize our approach of using PSNR in making path switching decisions to improve video quality.

Specifically, Fig. 7(b) shows a typical mapping from PSNR to video quality, as given by Eq. 16. The curve identifies three major “semi-linear” regions. Region I corresponds to a high loss factor region with low PSNR values, where the quality is consistently close to the worst possible score. Within this region, differences in PSNR do not translate into meaningful quality improvements. From a path switching perspective, two paths with such high loss factors are equally undesirable, in spite of their PSNR differences. In other words, even a positive PSNR difference should not trigger a path switching decision. A similar result, albeit for opposite reasons, holds in Region III that corresponds to paths with sufficiently low loss factors, so that the resulting PSNR values are all high enough to generate near perfect video quality. As a result, even when two paths in this region have a positive PSNR difference, switching between them will again not yield a meaningful improvement in video quality. The only region where differences in PSNR translate into substantial differences in video quality, and can therefore be used to effectively guide path switching decisions, is Region II. In this region, video quality improves almost linearly with PSNR, so that our earlier methodology for computing differences in PSNR from network measurements can be used to trigger path switching decisions that improve video quality.

Given the above analysis, the only remaining issue is to determine where the boundaries of the above three regions lie in terms of network parameters, and in particular loss factors. We discuss next a simple methodology that we derived and validated for that
4.3 Path switching and its evaluation

Because there is a cost associated with switching from one path to another even if its impact on quality can be successfully hidden, and because accurately estimating network performance calls for accumulating a sufficient number of samples, path switching decisions should be made at a relatively coarse time granularity. This granularity can range from a few tens of seconds to a few minutes, and in our experiments was set to 3 minutes. In addition, because the above region based approach is intrinsically an approximation, switching from one path to another should typically only be performed if “sufficient” quality improvement can be achieved by routing the video packets onto the alternative path. In our experiments, a PSNR difference of 1.5 dB was used as the threshold below which path switching should not be considered. To summarize, the path switching strategy we follow consists of the following steps:

- If either of the loss processes estimated on the two paths is in region I, i.e., $P^1_e > 1/(TTL)$ or $P^2_e > 1/(TTL)$, where $P^1_e$, $P^2_e$ stand for the loss event probabilities on path 1 and path 2, we choose path 1 over path 2 if and only if $P^1_e - P^2_e > 1/(TTL)$, and vice versa.
- If the loss processes on both paths are in region III, i.e., $P^1_e < 1/(TTL)$ and $P^2_e < 1/(TTL)$, no path switching needs to be performed.
- Otherwise, we choose path 1 over path 2 if and only if $\xi^2/\xi^1 > 1.4125$ or vice versa, i.e., the potential quality improvement should be greater than 1.5 dB.

The effectiveness of the above strategy was tested experimentally using a 3 minutes video clip Tonight Show, and two candidate paths connecting UPenn and UMN. The video packets are transmitted approximately every 10 ms. The experiment lasted for 10 hours during which the video clip was repeatedly transmitted on both paths. The transmission of the clip on both paths allowed the simultaneous evaluation of the quality of the received video over each of them (see Figs. 9(a)(b)). This was then used to assess the benefits of path switching against our baseline strategy that uses only one path at a time. In addition to the video stream, probing flows were also transmitted on both paths. Probes were generated every 40 ms, and the collected loss traces were fed to the loss estimation process that drove the path switching decisions. In particular, the loss parameters of each path were estimated every 3 minutes (4,500 probes), which was also the time granularity of path switching. The received video streams with path switching was “reconstructed” by piecing together the associated segments from the video streams received on each path, following the path switching decisions based on probing and quality difference estimations. The quality of the resulting stream was then evaluated. The results in Fig. 9(c) clearly illustrates the improvement in video quality achieved by path switching. Table 1 provides additional information on the benefits of path switching by comparing the average and the standard deviation of the quality score on each individual path to those of the video resulting from path switching. The statistics clearly show that our scheme not only improves the overall quality, but also reduces its variations.

Obviously, the effectiveness of path switching depends not only on the decision process itself, but also on the potential performance improvements offered by the paths. If all paths have similar loss characteristics, e.g., they share a common congestion point, switching paths is of limited use. Similarly, if the loss patterns on the monitored paths are transient or unpredictable, the accuracy of the loss estimation component will suffer, hence the effectiveness of path switching will also be limited. However, when loss patterns across candidate paths are uncorrelated, while at the same time losses on a path exhibit some level of temporal correlation, i.e.,

```
Table 1: Quality improvement from path switching

<table>
<thead>
<tr>
<th>Path 1</th>
<th>Path 2</th>
<th>Path switching</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overall quality</td>
<td>0.251</td>
<td>0.214</td>
</tr>
<tr>
<td>Quality variation</td>
<td>0.198</td>
<td>0.176</td>
</tr>
</tbody>
</table>
```

Note that although it is formulated for the case of two paths, it is readily extendable to the cases of more than two paths.
paths have non-overlapping and extended congestion periods, then path switching can be effective in improving network and application (video) performance. Two such instances can be seen on Fig. 9 at time 160 minutes (on path 2) and at time 400 minutes (on path 1). A comprehensive study of this issue is beyond the scope of this paper, and more discussions can be found in [16] demonstrating that such conditions are commonly encountered.

5. CONCLUSION

This paper investigated the use of dynamic path switching to improve the quality of streaming video, when it is transmitted over a network that offers path diversity. The paper focused on understanding how network level performance differences can be translated into improved video quality. This was motivated by the fact that in many cases a purely network based path switching decision can result in poorer video quality. Addressing this issue called for developing a simple yet reasonably accurate model for comparing the video quality achievable on different paths. The paper’s contributions were in developing a model based on which the best path, in terms of video quality, can be correctly selected in real-time, using only simple measurements of network performance. Although the model was initially used to drive path switching decisions, it obviously has broader applicability. For example, it can be used to test the readiness of a network in supporting streaming video applications, as well as to design quality-based adaptation schemes for video transmission.

There are several natural extensions for this work, and we briefly mention two of them that we are currently pursuing. The first is to extend the use of path switching to other applications such as VoIP and interactive video for which both losses and delay need to be taken into account. A second extension involves combining path switching with existing video adaptation schemes, e.g., layered coding, to further enhance the robustness of video transmissions over best-effort networks.

6. REFERENCES


Figure 9: Video quality improvement achieved by performance estimation/prediction and dynamic path selection: Quality variations (a) on path 1 (left), (b) on path 2 (middle), and (c) with dynamic path switching (right).

Equation-based congestion control for unicast applications.


Video quality measurement techniques. NTIA Report 02-392, June 2002.