Improving VoIP Quality Through Path Switching

Shu Tao, Kuai Xu, Antonio Estepa, Teng Fei
Lixin Gao, Roch Guérin, Jim Kurose, Don Towsley, Zhi-Li Zhang

Abstract— **The current best-effort Internet cannot readily provide the service guarantees that VoIP applications often require. Path switching can potentially address this problem without requiring new network mechanisms, simply by leveraging the robustness to performance variations available from connectivity options such as multi-homing and overlays. In this paper, we evaluate the effectiveness and benefits of path switching in improving the quality of VoIP applications, and demonstrate its feasibility through the design and implementation of a prototype gateway. We argue for an application-driven path switching system that accounts for both network path characteristics and application-specific factors (e.g., codec algorithms, playout buffering schemes). We also develop an application path quality estimator based on the ITU-T E-model for voice quality assessment, and an application-driven path switching algorithm that dynamically adapts the time scales over which path switching decisions are made to maximize voice quality. Through network emulation and experiments over a wide-area multi-homed testbed, we show that, with sufficient path diversity, path switching can yield meaningful improvements in voice quality. Hence by exploiting the inherent path diversity of the Internet, application-driven path switching is a viable option in providing quality-of-service to applications.**

I. Introduction

The consolidation of the Internet as the de facto communication infrastructure, the increasing availability of broadband access, and the emergence of integrated applications have all contributed to finally propelling Voice over IP (VoIP) to the forefront. However, VoIP like many other real-time applications requires minimum service guarantees that go beyond the best-effort structure of today's IP networks. In particular, even if modern codecs are capable of some level of adaptation and error concealment, VoIP quality remains sensitive to performance degradation in the network. Ensuring that the network is capable of meeting those service requirements is, therefore, key to the acceptance of VoIP as the technology of choice for voice transmission. There are two possible approaches for realizing this goal that are complementary but embody different perspectives on how to deliver service guarantees.

The first possibility involves adding service differentiation mechanisms to the network itself, and configuring them to dedicate resources and delivering service guarantees to VoIP traffic. The main disadvantage of this approach is in the added complexity it imposes on the network, and, more important, the difficulty of deploying it ubiquitously across a network of the size of the Internet. Consequently, network-based service differentiation has so far been most successful in limited settings, such as when controlling the allocation of bandwidth on an access link to ensure a sufficient share to VoIP traffic. The problem of delivering robust end-to-end service guarantees, therefore, remains and has motivated the development of other, more application-centric solutions.

In particular and in keeping with the original end-to-end principle of the Internet [1], another alternative is to rely on the *diversity* of connectivity options often available, and let applications decide how to best use them [2], [3], [4], [5]. Specifically, in this paper we explore the extent to which voice quality can be improved, when VoIP traffic is able to dynamically select from among several paths based on their performance. The expectation is that unless all paths are simultaneously experiencing poor performance, a path switching mechanism that consistently picks the best path should offer meaningful improvements in voice quality. Obviously, there are many potential obstacles to realizing such a goal. They range from the difficulty of predicting future path performance, to the challenge of mapping network performance to voice quality, to the added complexity of implementing the required functionality. In this paper, we carry out a comprehensive investigation aimed at demonstrating and quantifying the extent to which path diversity is capable of delivering strong performance guarantees to VoIP traffic.

Path switching as introduced in this paper is not a new concept, and has been previously investigated in a number of different settings. For instance, commercial solutions, such as [6], [7], monitor the quality of paths offered by different network providers and use that information to select the best provider. Similarly, the work of [8], [9] demonstrated the use of a similar principle in the context of overlay networks, where the detection of performance degradation triggers the use of different overlay paths. Our earlier work [10] illustrated the benefits of path switching in improving performance when both multiple providers and overlay paths were available. More recently and probably most relevant to this paper, Skype [11] introduced a similar idea in their peer-to-peer VoIP system, which maintains multiple connections for a single session and dynamically chooses the one that is best suited at the time. In this paper, we expand on these earlier works in two main directions. First, we explore the *application level* benefits that dynamic path switching affords to VoIP traffic, and develop a simple scheme for realizing those benefits. Second, we quantify the magnitude of those benefits in a variety of environments and demonstrate the *feasibility* of our proposed scheme through an operational prototype.

The paper demonstrates the feasibility of providing reasonably stable performance guarantees to VoIP traffic, simply by leveraging the path diversity that the Internet offers, and without changes to either the network itself or the application software clients. When coupled with a mechanism that estimates

Shu Tao and Roch Guérin are with the University of Pennsylvania; Kuai Xu and Zhi-Li Zhang are with the University of Minnesota; Antonio Estepa is with the University of Sevilla, Spain; Teng Fei, Lixin Gao, Jim Kurose, and Don Towsely are with the University of Massachusetts, Amherst.

the voice call quality supported by each path, our prototype gateway was found to deliver meaningful improvements across a broad range of network conditions. This was demonstrated using both a network emulator that allowed us to experiment with a variety of network impairments in a controlled fashion, and a wide-area testbed involving three sites connected by both multiple providers and several overlay paths. The prototype gateway is lightweight and capable of delivering voice quality that closely approximates its optimal feasible value that can only be achieved if assuming *a priori* knowledge of path quality variations and always selecting the best path at every instant. We believe that the use of simple techniques such as those described in this paper can help the current best-effort Internet accommodate the more stringent quality requirements of applications such as VoIP.

The remainder of this paper is organized as follows. Section II provides a high-level description of the architecture of our path switching system, and explains our design choices. Section III is devoted to a detailed discussion of how path quality is estimated. In Section IV, we study the various issues related to designing a path switching mechanism and developing a practical scheme that adapts to changes in path characteristics. Performance results are presented in Section V that demonstrates the effectiveness of path switching for a broad range of emulated environments and our wide-area testbed. Finally, Section VI summarizes our findings and points to several extensions we are currently considering.

II. Application-Driven Path Switching: A System Overview

Exploiting path diversity to improve the quality of applications such as VoIP raises several challenges. First, the perceived quality of a voice call depends on many factors, and not merely network factors such as packet delay or loss. Furthermore, the relation between network performance and voice quality is often non-trivial, determined partly by application-specific parameters such as codec algorithms, playout buffer size, error concealment mechanisms, etc. In other words, simply selecting the path with the best network performance is not adequate. This calls for an *application-driven* path switching system that takes into account both network performance and application-specific parameters. Such a system needs to i) measure and predict network performance of various paths, ii) map measured/predicted network performance into application quality estimates, and iii) make judicious path switching decisions based on the estimated application quality. Finally, implementation must be efficient and scalable in order to be deployable on an Internet scale, ideally without requiring modifications to the network or the applications.

The above observations and requirements motivate the design of *application-driven path switching* (APS) gateways, which reside at the edge of access networks, between end hosts and the larger Internet. The APS gateways perform dynamic path switching using application-specific quality estimation derived from measured network path performance. While path switching decisions are application-driven, the operations of these gateways are transparent to end hosts and applications.

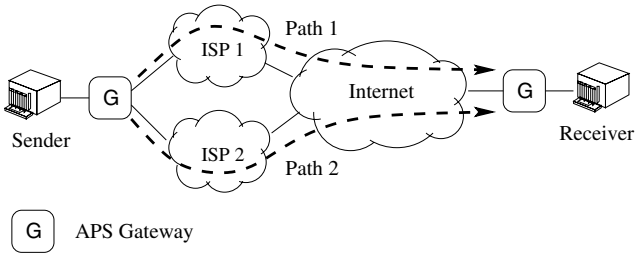


Fig. 1. A typical application scenario of our path switching scheme.

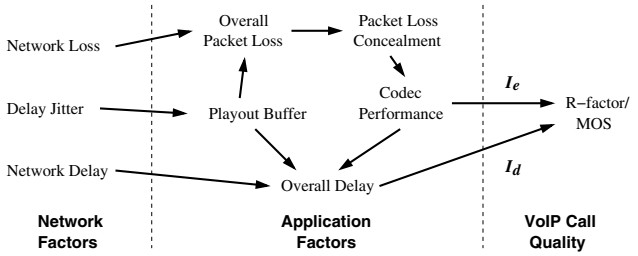


Fig. 2. Factors affecting overall loss and delay and their relations to voice quality.

A typical application scenario of our path switching system is shown in Fig. 1: a VoIP service provider has multiple paths available (e.g., by connecting to several ISP's) to carry the traffic from its customers. At each end of the communication, an APS gateway is used to dynamically determine which path should be used to forward voice packets. The goal of path switching is to maintain the best possible network conditions for voice data transfer so that the application quality perceived by a user can be improved. Before we describe the key components of our APS gateway, we first briefly discuss the various factors that affect VoIP call quality and describe how call quality is typically assessed. This provides the necessary background as well as justification for the design of our APS gateway.

A. VoIP Call Quality and Its Assessment

A typical VoIP application works as follows. First, the sampled and digitized voice signal is encoded using a given algorithm (e.g., G.711, G.723.1, G.729, etc.). Then, the encoded data (called frames) is packetized and transmitted using RTP/UDP/IP. At the receiver's side, data is de-packetized and forwarded to a playout buffer, which smooths out the delay jitter incurred by network transmission. Finally, the voice data is decoded and the reconstructed voice signal is delivered to the listener.

Packet loss and delay affect the perceived quality of a VoIP application in a complex manner. As shown in Fig. 2, factors affecting overall packet loss and delay come from both the network and the application. For instance, packets can be lost in the network or dropped by the playout buffer due to large delay jitter, while network delay, encoding and packetization delay, playout buffering delay, etc., all contribute to delay. For a path switching system the goal is obviously to minimize the negative impact caused by all network factors.

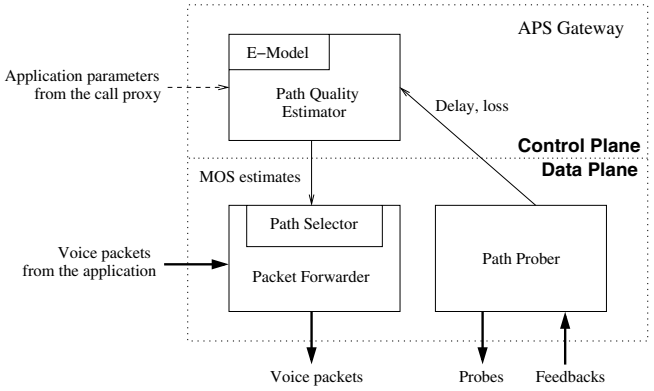


Fig. 3. The architecture of an application-driven path switching gateway.

Perceived voice quality is typically measured by the *mean opinion score* (MOS), a subjective quality score that ranges from 1 (unacceptable) to 5 (excellent). MOS values can be obtained by carrying out subjective tests [12]. Some objective models, e.g., PESQ [13], can also generate quality scores close to subjective results by comparing the impaired voice signal with its original version. However, neither can be used for real-time on-line quality assessment. In addition, objective evaluation methods like PESQ do not consider the effect of delay on voice conversations.

For application-driven path switching, we need a method for estimating call quality from the measured network performance such as delay and loss characteristics of a path. The ITU-T E-Model [14] provides a framework for this purpose. It defines an R-factor that combines different aspects of voice quality impairments:

$$R = R_0 - I_s - I_e - I_d + A, \tag{1}$$

where R_0 groups the effects of various noises; I_s includes the effect of other impairments that occur simultaneously with the voice signal; I_e covers the impairments caused by different types of losses; I_d represents the impairment caused by delay; finally, A compensates for the above impairments under various user conditions. The R-factor ranges from 0 to 100 and is related to MOS through a non-linear mapping [14]:

$$MOS = 1 + 0.035R + 7 \times 10^{-6} R(R-60)(100-R). \tag{2}$$

Among all of the factors in Eq. (1), only I_d and I_e are typically considered variable in a VoIP system. Using default values for all other factors reduces the model to [15]

$$R = 94.2 - I_e - I_d. \tag{3}$$

It can be seen from Fig. 2 that the mapping between network loss and delay and I_e and I_d, respectively, involves several application factors, such as packet loss concealment (PLC) and codec performance. It is therefore infeasible to determine the application quality of a path simply based on network loss and delay measurements. In other words, path selection depends not only on network loss and delay characteristics, but also on application level factors.

B. APS Gateway Architecture

We are now in a position to describe the APS gateway architecture, whose block diagram representation is shown in Fig. 3. It includes the following key components.

1) Path Prober: The path prober periodically sends UDP probes on each available path. Probes are generated to emulate the behavior of a VoIP voice call. The APS gateway on the receiver side returns the delay and loss statistics for these probes that are needed by the E-Model. In our experiment, the size of a probe is 50 bytes, and the probing interval is configured as 20 ms. Since we only generate one probing flow along each path, and the results of probing are usable by all VoIP calls using the same path, the associated overhead is relatively small. All APS gateways have synchronized system clocks (via CDMA clock synchronization devices), so that accurate one-way delay measures can be obtained from the time stamp of each probe.

2) Application Path Quality Estimator: The application path quality estimator translates network path measurements (packet loss rate and delay obtained from the path prober) into application quality estimates using the E-model. This translation is based on Eqs. (2) and (3), and involves deriving expressions for I_d and I_e according to application-specific parameters, such as codec algorithms and playout buffering schemes. The derivation of those expressions is carried out off-line for all possible combinations of commonly used voice codecs (e.g., G.711, G.729, G.723, etc.), PLC mechanisms (e.g., silence substitution, waveform substitution, etc. [16]), and playout buffering schemes (static or adaptive). In Section III, we illustrate this process for a few typical configurations. Note that the application parameters can be obtained from the call proxy, e.g., SIP proxy server or H.323 gatekeeper, which is used in VoIP systems for session initiation and is therefore aware of the configurations on both sides of a VoIP call. Once the call proxy sends the required information to the APS gateway, the gateway can match them to its predefined mappings and find the correct one to use for each voice call.

3) Packet Forwarder: The packet forwarder is integrated with the path switching mechanism (path selector), which dynamically decides the best performing path for each voice call[1]. Once a path is selected, the packet forwarder forwards the voice packets along the corresponding path. In our system, each APS gateway is assigned multiple IP addresses. BGP routing policies are configured at the border gateway routers such that packets with a given source or destination address are delivered through one of the provider ISP's [10]. Thus, in order to forward packets along a selected path, the packet forwarder simply encapsulates the voice packets with the corresponding source or destination address and sends them to the receiver's APS gateway. At the receiver side, the APS gateway decapsulates these packets and further forwards them to the receiver.

[1]Here we assume that APS gateways maintain per-call state and perform path switching decisions for each call. It is possible to perform application path switching on an aggregate basis to increase scalability, for example, by grouping voice calls with the same set of application parameters and destination.

TABLE I

THE VALUES OF γ_1, γ_2 AND γ_3 CALIBRATED IN [17].

Codec	frames/pkt	PLC	γ_1	γ_2	γ_3
G.723.1.B-5.3	1	silence	19	71.38	6
G.723.1.B-6.3	1	silence	15	90.00	5
G.729	1	silence	10	47.82	18
G.723.1.A+VAD-6.3	1	none	15	30.50	17
G.729A+VAD	2	none	11	30.00	16

III. APPLICATION PATH QUALITY ESTIMATION

In this section, we describe the design of our application path quality estimator module, with a focus on deriving expressions for the parameters I_e and I_d of the E-model that incorporate both application-specific characteristics and network measurements.

A. Estimating the Impact of Loss

I_e accounts for impairments caused by both encoding and transmission losses. For a given codec, encoding loss is a predetermined value. However, due to the use of various PLC mechanisms and the performance difference of voice codecs, the mapping from loss to I_e is not straightforward. It has been proved in several tests [17], [15] that the relation between I_e and the overall packet loss rate e can be expressed as follows:

$$I_e = \gamma_1 + \gamma_2 \ln(1 + \gamma_3 e), \tag{4}$$

where γ_1 is a constant that determines voice quality impairment caused by encoding, and γ_2 and γ_3 describe the impact of loss on perceived voice quality for a given codec. Note that e includes both network losses and playout buffer losses.

A typical approach to obtain the values of γ_1, γ_2, and γ_3 is to use voice quality testing results, i.e., to evaluate the quality of a voice signal (using subjective or objective methods) in different loss conditions. In [17], a simulation method is introduced to calibrate γ_1, γ_2 and γ_3. Table I summarizes the results of [17] for several codecs. The same approach can be applied to obtain γ_1, γ_2, and γ_3 for other codecs not listed in Table I. In this paper, we focus on G.729A as the reference codec and use the parameters provided in Table I to estimate the impact of loss on voice quality for a given path.

B. Estimating the Impact of Delay

The end-to-end delay, d, determines the interactivity of voice communication. Its impact on voice quality depends on a critical value [15]: when end-to-end delay exceeds 177.3 ms, voice quality degrades more rapidly. This effect can be modeled [15] in the form of

$$I_d = 0.024d + 0.11(d - 177.3)\mathbf{I}(d - 177.3), \tag{5}$$

where $\mathbf{I}(x)$ is an indicator function: $\mathbf{I}(x) = 0$ if $x < 0$, and 1 otherwise.

In a typical VoIP system, d is mainly composed of

- network delay ($D_{network}$), including propagation delay and queueing delay – the latter is the major cause of delay jitter;
- codec-related delay (D_{codec}), including the delay incurred by packetization and encoding (look-ahead and processing);
- playout buffering delay ($D_{playout}$), which is the delay incurred by the playout buffer at the receiver to absorb the jitter introduced during data transfer.

TABLE II

CODEC-RELATED DELAY FOR DIFFERENT CODECS.

Codec	frame size	frames/pkt	packetization	encoding	D_{codec}
G.723	30 ms	1	30 ms	17.5 ms	47.5 ms
G.729	10 ms	2	20 ms	15 ms	35 ms

The network delay, $D_{network}$, is measured by our Path Prober. The other two components, D_{codec} and $D_{playout}$, are associated with application configurations. For example, as shown in Table II, we encapsulate two G.729 frames into one RTP packet, which results in a packetization delay of 20 ms (derived from frames/packet $\times$ frame size). Furthermore, a 5 ms look-ahead delay and a 10 ms processing delay are introduced in the encoding process. Thus, we estimate D_{codec} for G.729 as 35 ms.

$D_{playout}$ denotes the delay introduced by the playout buffer at the receiver, which helps smooth out the delay jitter introduced in the network. Its value is specific to the implementation of the playout buffer. For VoIP applications, both static and dynamic playout buffering schemes are commonly used. In a static scheme, $D_{playout}$ is chosen close to the maximum (e.g., 99% quantile) of the variable part of the end-to-end delay. Meanwhile, the buffer size is usually configured as twice the number of packets that can be generated in $D_{playout}$ to avoid buffer overflow. In this case, the value of $D_{playout}$ can be obtained from the call proxy (a SIP proxy server or H.323 gate-keeper) as a configuration of the voice receiver. In a dynamic buffering scheme, the receiver can adapt the size of its playout buffer (and $D_{playout}$) according to the change of the filling level in the playout buffer. In our analysis, we assume that the receiver can adaptively choose the size of its playout buffer, so that buffer overflow never occurs even when packets arrive earlier than expected. To deal with late packets, the decoder needs to consider the following trade-off in selecting $D_{playout}$. On one hand, if $D_{playout}$ is too small, buffer underflow (and therefore playout interruption) can occur when packets arrive late (which is equivalent to a loss). On the other hand, as $D_{playout}$ increases, the probability of buffer underflow decreases but the overall end-to-end delay increases. To illustrate the above trade-off, we plot the packet arrival and decoding curves seen by the playout buffer, in Fig. 4. Without considering network loss, the average packet arrival rate (r) equals the average decoding rate. Let $A(t)$ denote the cumulative packet arrival function (i.e., the number of packets received during $[0, t]$), and $C(t) = r(t - D_{playout})$ the curve for packets consumed by the decoder. Thus, the probability of playout buffer underflow is

$$E_{playout} = \Pr\{A(t) - r(t - D_{playout}) < 0\}. \tag{6}$$

Note that t represents the time that the first packet arrives at the receiver's playout buffer. The total end-to-end loss probability

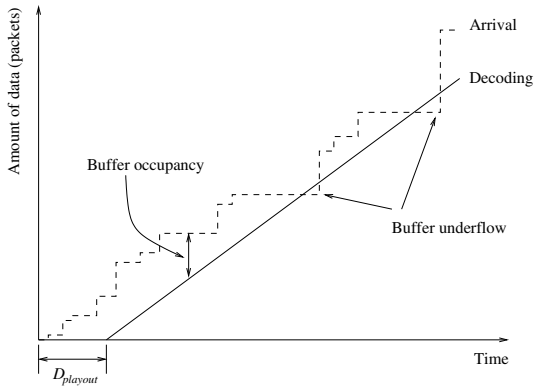


Fig. 4. The relation between loss, delay, and playout buffering.

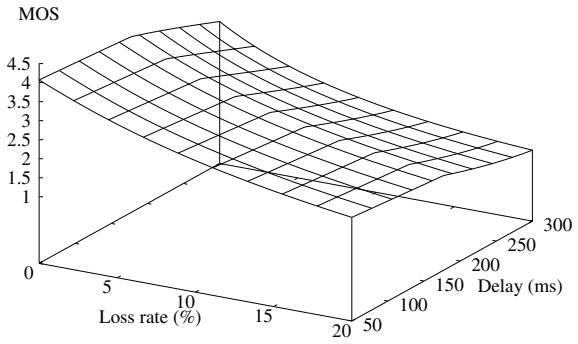


Fig. 5. VoIP quality as a function of end-to-end loss rate and delay.

is the sum of network loss probability $E_{network}$ and buffer underflow probability $E_{playout}$, i.e.,

$$e = E_{network} + E_{playout}. \tag{7}$$

While the end-to-end delay is

$$d = D_{network} + D_{codec} + D_{playout}. \tag{8}$$

C. Summary

Based on the above analysis, we can derive the end-to-end delay d and loss probability e by measuring the average loss rate ($E_{network}$), network delay ($D_{network}$), and by estimating the playout buffering loss ($E_{playout}$), the encoding and packetization delay (D_{codec}) and the playout buffering delay ($D_{playout}$). Combining Eqs. (8) and (5), we can derive the delay factor I_d from $D_{network}$, D_{codec}, and $D_{playout}$. Also, using Eqs. (7) and (4) we can obtain the loss factor I_e from $E_{network}$ and $E_{playout}$. When the receiver is using a dynamic playout buffering scheme, the value of $D_{playout}$ is not known *a priori*, and both I_e and I_d are functions of $D_{playout}$. Hence, the overall R-factor is also a function of $D_{playout}$. We use the optimal playout delay $D^{opt}_{playout}$, which maximizes the value of R to, approximate $D_{playout}$. Namely,

$$D^{opt}_{playout} = \arg\max R(D_{playout}). \tag{9}$$

By doing so, the quality of a path can still be estimated without knowledge of the voice receiver's playout buffering scheme[2]. Given I_d and I_e, we can calculate the R-factor using Eq. (3), and from Eq. (2), a MOS score can be assigned as the estimated application quality of a path.

To illustrate that the MOS score can relate to end-to-end delay and loss in a nontrivial non-linear fashion, we plot in Fig. 5 the MOS score of a path as a function of both end-to-end loss rate and delay using the G.729A+VAD codec. We see that the same delay affects voice quality differently for different values of loss rates, and vice versa. Overall, losses have a more significant impact on voice quality than end-to-end delay. As pointed out, the mapping is application-specific, and using a different codec with different playout buffering and error concealment schemes will yield a different relation between loss, delay, and quality.

[2]Note that the optimal playout buffering delay can be difficult to realize for a real voice receiver, since it requires an accurate prediction of the delay/jitter statistics on the path. However, it can serve as a good reference value to estimate the best possible call quality supported by a path.

IV. Application-Driven Path Switching

In this section, we discuss the key issues in the design of a practical application-driven path switching mechanism and describe the path switching algorithm used in the APS gateway. A key feature of our algorithm is that it dynamically adapts the time scale at which switching is made to maximize the gain of path switching in improving application quality.

A. Path Quality Prediction

The estimated quality of a path is based on past path performance, while path switching requires predicting the future quality of a path. Previous studies [18], [10] show that in predicting the performance of a path, sophisticated prediction models do not outperform simple models in most cases. Therefore, we use a simple predictor in our design.

Let Q_i represent the voice quality (MOS) of path i. We evaluate the quality of a path in time slots of length t and denote the quality of path i in time slot k as $Q_i^{(k)}$. We use the path quality predictor $Q_i^{(k)} = Q_i^{(k+1)}$. Obviously, the accuracy of such a prediction depends on how rapidly path quality varies and on the value of t, and path switching decisions based on this prediction may not be always beneficial. We describe next a method to characterize path quality variations and perform path switching only when it is likely to be beneficial.

B. Estimating the Benefits of Path Switching

The quality improvement achievable through path switching is not determined by the path switching mechanism alone. The characteristics of the candidate paths are important factors that affect performance. The best candidate paths for our path switching scheme should satisfy the following requirements:

- Performance variations on the paths are independent of each other so that when one path is experiencing

performance degradations, it is unlikely that the other paths have the same problem.
- Performance fluctuations on each path exhibit temporal correlations, i.e., path quality should be predictable so that the path switching decisions based upon quality prediction are meaningful.

Our previous study [10] has shown that in many cases, diverse paths obtained from either multi-homing or overlay networks have relatively independent performance variations. However, this need not hold for all paths and even for a given set of paths the levels of performance correlations may change over time. It is, therefore, important to continuously assess the potential benefits of path switching based on path performance prediction, as well as determine over what time scale, t, path switching would yield the most benefit so that path switching decisions can be dynamically tuned to maximize their effectiveness.

We develop the following methodology and metrics to quantify the potential benefit of path switching over a certain period of time. Assume that we have a set of m candidate paths, whose qualities have been monitored over T time slots (each of length t). We have two goals. First, we want to establish a baseline reference against which to compare the benefits of path switching. Second, we want to devise a mechanism to determine the time scale at which to perform path switching. For the purpose of our first goal, we assume perfect knowledge of path quality over the past T time slots and denote r as the path with the *best overall quality*. We will also refer to r as the baseline reference. Next, we compare the voice quality of using path r to that achievable using "ideal" path switching, i.e., when making perfect switching decisions to always pick the best performing path in the next time slot. Assuming that this is path s in slot k, the gain in the voice call quality at time slot k is $\rho_k = Q_s^{(k)} - Q_r^{(k)}$, and the overall gain of ideal path switching is

$$\rho = \frac{1}{T}\sum_{k=1}^{T}\rho_k. \tag{10}$$

In practice, however, we do not know the quality of paths in the next time slot. Hence, we have to rely on predicted quality to make path switching decisions. Since we use $Q_i^{(k)}$ to predict $Q_i^{(k+1)}$, the *actual* gain of prediction-based path switching in time slot k is $\hat{\rho}_k = Q_{\hat{s}}^{(k)} - Q_r^{(k)}$, where $\hat{s} = \arg\max_i Q_i^{(k-1)}$, i.e., the path with the best quality in the previous time slot $k-1$, and the overall gain of prediction-based path switching is

$$\hat{\rho} = \frac{1}{T}\sum_{k=1}^{T}\hat{\rho}_k. \tag{11}$$

Define $\alpha = \hat{\rho}/\rho$. The pair of parameters (ρ, α) captures the benefit of path switching and how much of that benefit is feasible based on prediction. If ρ is small, then there is little benefit of path switching to begin with. If ρ is large but α is small, there is potential benefit in path switching, but it is difficult to achieve using the simple predictor. The latter is typically caused by constant path quality variations from one time slot to the next. Hence the time scale t over which we estimate/predict path quality and make path switching decisions also plays an important role in the performance of a path switching mechanism[3]. There is clearly a trade-off in the choice of t: a small value of t allows rapid reactions to quality variations among the candidate paths and thus maximizes the (ideal) gain in path switching ρ; on the other hand, a small value of t is more likely to cause prediction errors when network performance fluctuates rapidly, therefore reducing the actual gain $\hat{\rho}$ of prediction-based path switching. In the next section, we present an adaptive path switching algorithm that evaluates path qualities over a range of time scales and dynamically selects the best time scale for making path switching decisions.

C. Time-Scale Adaptive Path Switching Algorithm

In our design, we adaptively select t among a set of predefined values $\{t_n; n = 1, 2, ..., N\}$. For each value of t_n, we keep track of the quality improvement that would have been achieved in the past if t_n had been used as the time window for quality prediction and path switching. If $\hat{\rho}_k(t_n)$ represents the gain of using $t = t_n$ (in time slot k that has length t_n), we need to dynamically update the estimate of $\hat{\rho}(t_n)$ every t_n seconds. This is achieved by using an exponentially-weighted moving average $\tilde{\rho}(t_n)$[4], i.e.,

$$\tilde{\rho}_k(t_n) = \tau\hat{\rho}_k(t_n) + (1-\tau)\tilde{\rho}_{k-1}(t_n), \tag{12}$$

where $\tau = 0.2$ (we tried different values, and 0.2 is selected in order to avoid oscillations during the adaptation process, while maintaining the responsiveness of the algorithm) . In computing $\hat{\rho}_k(t_n)$, we use the path, say, path $\hat{r}$ which has the best overall quality *up to the current time* k as the reference[5]. In our path switching mechanism (see Algorithm 1), we define the set of time slot length as $t_n = 15 \times 2^{n-1}$ seconds for $n \in \{1, 2, 3, 4, 5, 6\}$. The algorithm runs a clock with 15 seconds as its basic time unit. At the end of each time unit, $\tilde{\rho}(t_n)$ will be updated. The algorithm adjusts the path switching time scale to ensure that we select the one that maximizes the estimated quality improvement, i.e., the algorithm adaptively selects the best value of t as $t = \arg\max\{\tilde{\rho}(t_n)\}$.

Once the time scale for path switching, t, is determined, the algorithm compares the predicted quality of the current path Q_s (measured over time scale t) with those of the other paths. If there exists another path i whose quality Q_i is better than Q_s by at least $\Delta = 0.1$, path i will be selected for the next time unit; otherwise, the algorithm continues to use the current path s. Here, using a relatively large Δ ensures that path switching is performed only if a sufficient quality improvement is expected.

[3]Note that both the ideal gain ρ and the predicted gain $\hat{\rho}$ are a function of t.

[4]The frequencies of updating different $\tilde{\rho}(t_n)$'s are not the same. At the end of each time unit, $\tilde{\rho}(t_n)$ is updated only if there has been t_n seconds since its last update.

[5]The overall quality of a path is also estimated by using a moving average similar to Eq.(12).

Algorithm 1 Path switching

```
1: for each unit of time do
2:    for n = 1 to N do
3:       update ρ̃(t_n)
4:    end for
5:    t ⇐ arg max{ρ̃(t_n)}
6:    if clock = t and ρ̃(t) > 0 then
7:       if Q_i − Q_s > Δ, i ≠ s then
8:          s ⇐ i, start using path s
9:       end if
10:      clock ⇐ 0
11:   end if
12:   clock ⇐ clock + 1
13: end for
```

D. On the Potential Negative Impact of Path Switching

Although path switching can potentially improve application quality, it can also possibly introduce performance degradations. For instance, in addition to the overhead incurred by path probing and quality estimation, switching from one path to another may also cause transient disruptions, especially when the candidate paths differ significantly in their propagation delay as this can result in an abrupt change in end-to-end delay. We argue that given the relatively low frequency of path switching and the expected range of delay increase or decrease, these differences can be readily absorbed by most playback mechanisms.

Consider a constant rate voice flow is switched from path A to path B. Consider first the case where the end-to-end delay on path A (d_A) is smaller than on path B (d_B). The switch causes a reception "gap" at the receiver of duration $d_B - d_A$ time, which can potentially lead to a discontinuity in voice playout at the receiver. Conversely, when $d_A > d_B$, the receiver will get a burst of out-of-order packets. Let r be the packet rate of the voice flow, if the path switching decision was made between the transmissions of packets k and $k+1$, then packets $k+1$ to $k+r(d_A-d_B)$ will arrive at the receiver before packet k, creating a burst of $r(d_A - d_B) + 1$ packets over a time interval of duration $1/r$ units of time.

However, the use of a playout buffer together with adaptive playout control can mitigate negative effects in both above scenarios [19]. Specifically, when path switching causes a sudden decrease of the buffer filling level, the receiver can adaptively reduce the decoding rate to avoid buffer underflow. Conversely, if path switching causes a sudden increase of the buffer filling level, the receiver can speed up the decoding rate to maintain the playout delay close to the optimal value. Since voice streams are typically composed of interleaved talk and silence spurts, such adaptation can be easily implemented by prolonging or shortening the silence spurt following the talk spurt affected by path switching[6].

[6]To illustrate the effectiveness of adaptive playout control, we generated speech samples that were subjected to both increases and decreases in end-to-end delay. More details can be found on *http://www.cs.umn.edu/research/networking/itrtestbed/*.

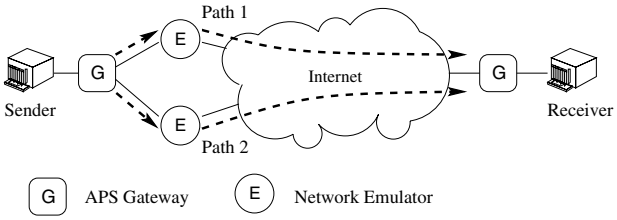


Fig. 6. A typical setting of using our network emulator to evaluate the performance of path switching.

V. Performance Evaluation

In this section, we verify the design of our path switching system and evaluate its performance. We conduct experiments both in an emulated environment and on a wide-area testbed. With emulation, we test the robustness of our path switching mechanism in various network conditions. The experimental results collected from the testbed are used to further verify the effectiveness and performance benefits of path switching. In order to evaluate the improvement in voice quality available from both an "ideal" path switching mechanism, i.e., one that assumes perfect knowledge of the future quality of all paths, and what we are able to accomplish using path probing together with our path switching mechanism, we initiate probing and voice flows simultaneously on *all* available paths. The traces consist of time-stamped received packets recorded for all combinations of paths. In other words, we record probe traces that enable us to recreate the results of our quality estimation procedure for each path, and also record the packet traces of voice flows sent over every possible path. The latter allows us to reconstruct off-line the composite packet trace that would have resulted from any path switching scenario, and therefore perform the kind of performance comparisons we are seeking.

A. Network Emulations

The network emulator used in our experiments acts as an extra node on the path (similar to that of [20]). When a packet reaches the emulator, it is delayed or dropped according to predefined models so that different delay/loss processes can be observed by the end-points. In our experiments, we generate delays from Pareto distributions with different means and variances and set a maximum value for the queueing delay in the emulated node, i.e., packets assigned delays larger than this value are dropped. A typical setting for the emulation is shown in Fig. 6.

1) Quality-based vs. loss-based path switching: Our path switching involves voice quality estimation, in which loss and delay are translated into voice quality using the E-Model. In this experiment, we demonstrate why such translation is necessary. We compare two path switching mechanisms: one triggered using voice quality estimates and the other triggered by loss rate estimates. The choice of loss rate was based on the fact that (discussed in Section III) losses are the dominant factor contributing to voice quality degradation, and proposing a parameter combining both loss and delay would have amounted to trying to reproduce the rationale behind the

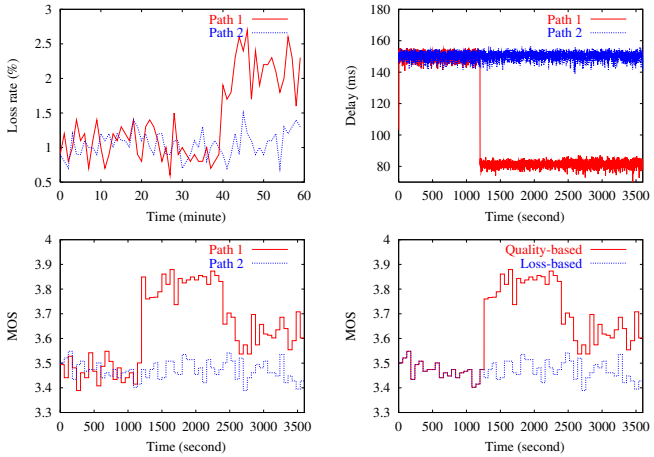


Fig. 7. (a) Loss rate (every min) variations on both paths (top left); (b) Delay variations on both paths (top right); (c) Quality variations on both paths (bottom left); (d) Resulting quality of quality-based and loss-based path switching (bottom right).

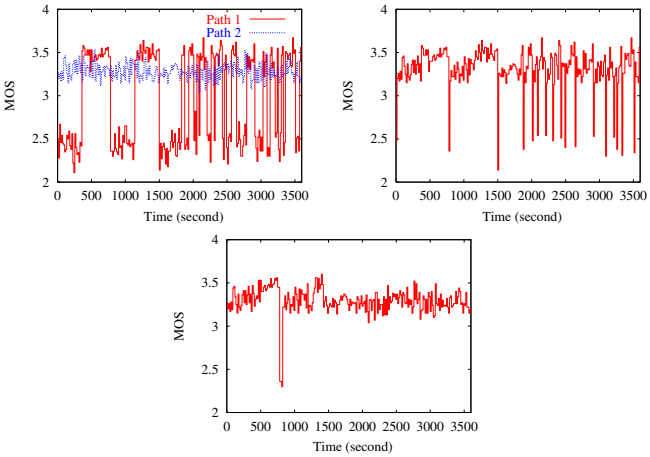


Fig. 8. (a) Quality variations on both paths (top left); (b) Resulting quality using fixed path-switching time scale ($t = 15$ secs) (top right); (c) Resulting quality using adaptive path-switching time scale (bottom).

development of the E-model. The experiment involves only two candidate paths that exhibit different emulated delay and loss characteristics. Specifically, both paths have packet loss rates around 1% at the beginning, while 40 minutes into the experiment the loss rate on path 1 increases to around 2%, as shown in Fig. 7(a). The average delays on both paths are about 150 ms at the beginning, and 20 minutes into the experiment the delay of path 1 drops to about 80 ms, as shown in Fig. 7(b). In a real network setting, such large delay changes can often arise because of AS-level path changes caused by BGP dynamics [10]. As a result of the changes, the quality of path 1 (evaluated every minute through its MOS value) begins with a MOS value around 3.5, jumps to around 3.8 at 20 minutes when its delay decreases from 150 ms to 80 ms, and drops back down to about 3.6 at 40 minutes when its loss rate increases to 2%. Throughout the experiment, the quality of path 2 remains around 3.5, as shown in Fig. 7(a).

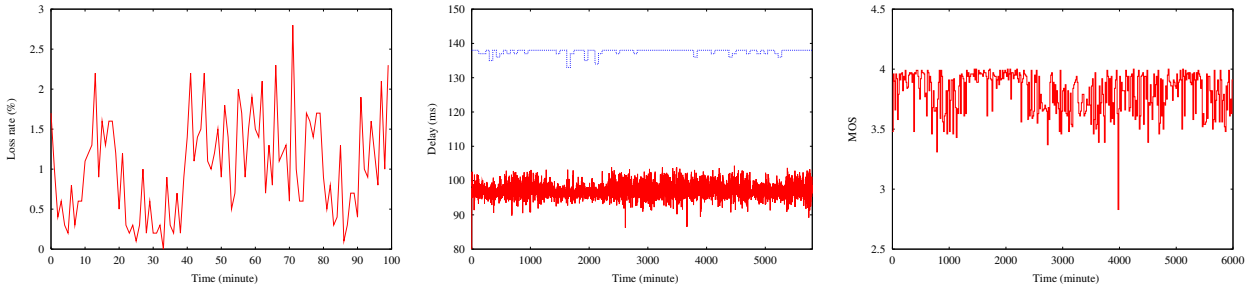
Fig. 7(d) clearly illustrates, for this particular scenario, the benefits of a quality-based path switching mechanism versus one that relies only on losses. Specifically, as can be seen in Fig. 7(c), the delay decrease that path 1 experiences 20 minutes into the experiment translates into a quality improvement that not only makes path 1 a better choice when both path 1 and path 2 have similar loss characteristics, but is also sufficient to offset the increased loss rate that path 1 starts displaying 40 minutes after the start of the experiment. Using loss rate as the only criterion would preclude switching to the better path in either one of those instances. This simple example illustrates the importance of using both delay and loss characteristics in making path switching decisions, and in particular using a methodology such as that of the E-model for translating them into voice quality measures that allow meaningful path comparisons.

2) Fixed vs. adaptive path-switching time scale: As discussed earlier, another important factor that can affect the effectiveness of path switching is the time scale used in estimating path quality and making path switching decisions. In order to track changes in temporal correlation in path quality variations, we adaptively adjust the value of the time interval t used in making path switching decisions, so that they remain as effective as possible. In the next experiment, we demonstrate the benefits of adapting the time-scale in making path switching decisions.

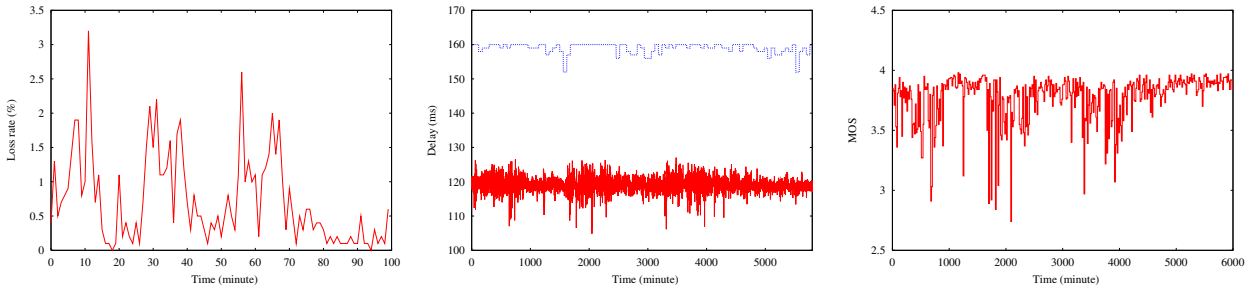
Fig. 8(a) shows the quality variations on two candidate paths: the quality of path 1 shows strong temporal correlation in the first 30 minutes of the experiments, with alternating periods of low quality, i.e., MOS values around 2.5, and periods of substantially higher quality, i.e., MOS values around 3.5. This is then followed by a period of much greater instability with rapid fluctuations in quality, often at a time granularity of less than a minute. Those rapid changes in quality make path switching decisions difficult, as they make predicting the (future) quality of path 1 difficult. In contrast, the quality of path 2 is relatively stable and remains at a MOS value around 3.2.

The difference between a fixed time scale of $t = 15$ seconds versus an adaptive time scale as described in Section IV is demonstrated in Figs. 8(b) and 8(c), respectively. The two figures show essentially similar voice quality during the first 30 minutes of the experiment, but very different results thereafter. In particular, Fig. 8(b) illustrates the many bad decisions that a fixed prediction interval produces, because of its inability to properly predict path quality in the presence of rapid fluctuations. In contrast, the adaptive scheme "learns" that the quality of path 1 is now more variable and increases the length of its decision interval as a consequence. This provides better and more stable performance, as shown in Fig. 8(c).

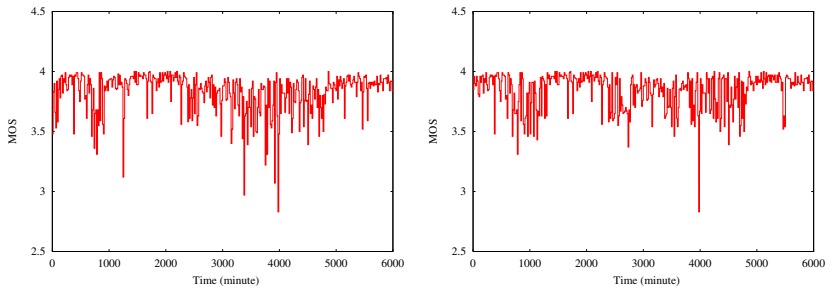
3) A more realistic example: For this last emulation experiment, we chose a scenario with delay and loss variations that are somewhat more representative of conditions that may arise in real network settings. We consider again two distinct paths with loss, delay and quality variations as shown in Figs. 9(a) and 9(b). In the figures, loss rates are one minute averages, delays are averaged over one second intervals (since voice samples are packetized every 20 ms, this corresponds to an average delay over 50 packets), and quality is computed every



(a) Loss rate (averaged every min), delay (averaged every sec), and quality (evaluated every 15 secs) variations on path 1



(b) Loss rate (averaged every min), delay (averaged every sec), and quality (evaluated every 15 secs) variations on path 2



(c) Resulting quality (evaluated every 15 secs) of fixed and adaptive time scale

Fig. 9. A more realistic example that demonstrates the effectiveness of adaptive path switching.

15 seconds. Note that the delay values include not only the network delay, but also the optimal playout buffering delay (about 40 ms in most cases). In addition, there is a 22 ms difference in propagation delay between the two paths. The overall average MOS values of path 1 and path 2 are 3.70 and 3.77, respectively, while the corresponding standard deviations are 0.212 and 0.202.

Fig. 9(c) gives the resulting voice quality when path switching is used with either a fixed time scale of 15 seconds or an adaptive time scale. Although the differences are not as dramatic as those of Fig. 8, the adaptive time scale scheme can again be seen to have an edge over the one using fixed time scale. Specifically, path switching based on an adaptive time scale yields an overall average MOS value of 3.91 with a standard deviation of 0.145, while using a fixed time scale results in comparable values of 3.83 and 0.170, respectively. The figure shows more graphically that an adaptive time scale is indeed able to avoid a number of bad switching decisions that a fixed time scale scheme makes, and also results in a lesser variability in voice quality.

B. Testbed Experiments

To further test and validate the benefits of our path switching scheme in improving voice quality, we set up a wide-area testbed consisting of three multi-homed campuses in the US, two on the East coast, University of Massachusetts (UMass) and University of Pennsylvania (UPenn), and one in the Midwest, University of Minnesota (UMN). Each campus can select from among different network providers to reach the other campuses, and overlay paths using the third campus as a transit node are also available between each pair of campuses. The testbed is shown in Fig. 10, where the connectivity available at each campus is as follows: (AB) corresponds to the default Internet connection through the Abilene network

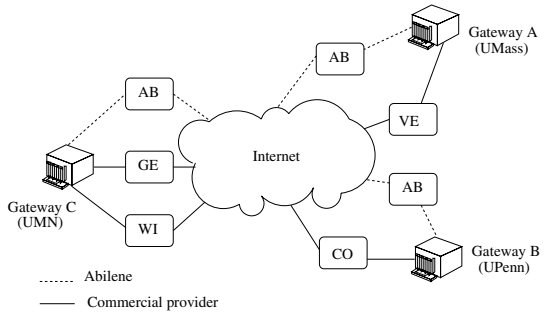


Fig. 10. The APS gateways in the testbed and their connectivities to different ISP's.

and is accessible from all three sites; in addition, each site can select from among one or more commercial service providers: UMass via Verio (VE), UPenn via Cogent (CO), and UMN via Genuity (GE) and Wiltel (WI), all of which are so-called tier-1 or tier-2 providers.

The application gateways are not themselves directly connected to the different ISP's, and rely instead on the existing campus routing infrastructure to access them. The ability to allow our application gateways to automatically select a specific outgoing ISP is based on the assignment of multiple IP addresses to the gateways, that are interpreted using special routing policies at the border routers of each campus to enforce the desired packet forwarding decisions. Specifically, the border routers at UPenn and UMN use source-address based routing, and the application gateways on the testbed at UPenn/UMN select an outgoing ISP's by choosing an appropriate IP address as their source address. UMass installs static routes to the other two sites in its border gateway, which selects an outgoing ISP based on the packets destination address. The previously mentioned additional overlay paths between campuses are implemented directly within the application gateway themselves using IP tunnels.

During a period starting on 05/22/2004 and ending on 06/15/2004, we collected 526 hours of traces on the paths connecting the three campuses. Table III summarizes some of the statistics derived from analyzing the traces. For each source-destination pair, we give the results collected on two candidate paths that were the most qualified for path switching. A path is identified using the notation *x-y(a)* or *x-y(a)-z(b)*, where the former denotes a direct path, while the latter corresponds to a two-hop overlay path. *x*, *y*, *z* represent the gateways in the testbed, and *a* and *b* represent the ISP used to reach a particular node. For instance, UMass-UPenn(AB)-UMN(CO) represents the path from UMass to UMN via UPenn with the network path via Abilene being used for the gateway at UMass to reach the gateway at UPenn, and the network path via Cogent being used for the gateway at UPenn to reach the gateway at UMN.

In order to assess the benefits of path switching in the context of this testbed, we first selected as a reference for each source-destination pair the path that provided the best voice quality over the duration of the experiment. In other words, this identifies the best baseline performance available without path switching. Because the Abilene network, one of the available connectivity choices between the different sites, is typically lightly loaded, our experiments showed that the best path offered very good voice quality most of the time. However, performance degradations do occur from time to time even on the best paths, and path switching is effective in maintaining voice quality during those periods. To highlight the quality gain achievable through path switching, we only consider time periods during which the reference path was experiencing some level of quality degradations (denoted as *duration* in Table III), i.e., when the MOS value of the reference path is lower than an alternative path by at least 0.1. Note that compared with the total duration of the experiments, the durations of such periods may appear to be short. However, path quality degradations can be expected to be more frequent in less favorable environments. We then computed the average gain in voice quality for both ideal path switching (*ideal gain*) and our prediction-based path switching (*actual gain*) during those periods.

It can be seen from Table III that our path switching scheme was effective at delivering meaningful improvements in voice quality for each of the source-destination pairs on our testbed for periods of at least 10 minutes. For some source-destination pairs, e.g., the UMass-UPenn pair, it was effective for more than 2.5 hours while our experiments ran. The minimum *actual gain* (in terms of MOS) we were able to achieve was 0.08 for the UMN-UPenn paths, while this value was as high as 0.83 for the UPenn-UMN paths, which represents a significant voice quality improvement. In addition, it is also worth noting that for all path pairs, the accuracy of prediction is quite high (no less than 62%). This suggests that path switching based on our simple adaptive quality predictor can yield improvements close to the ideal value. This is probably due in part to the relatively long duration of congestion events on most of the paths, which makes them more predictable, and therefore minimizes the number of instances that we lag behind the ideal path switching mechanism. Specifically, when the predictor detects a quality degradation on the current best path a quality improvement on another path, this situation is likely to last for a long enough period of time. As a result, the path switching decision we make will remain valid in multiple subsequent time slots. And, the penalty incurred by the predictor (compared to the ideal path switching that foretells the change one time slot earlier) is amortized over these time slots. An illustration of this behavior is shown in Fig. 11 based on traces collected on two overlay paths, UMN-UPenn(AB)-UMass(AB) and UMN-UPenn(AB)-UMass(CO), between 6:39pm and 7:39pm, 05/24/2004. Fig. 11(a) plots voice quality over the two paths during that period of time, and illustrates the long lasting nature of quality variations on both of them. Fig. 11(b) illustrates that as a result path switching is able to limit periods of degraded voice quality to of the order of a minute, i.e., the length of a decision interval, instead of over 30 minutes in the absence of path switching.

VI. Conclusions and Future Work

In this paper, we evaluated the effectiveness and benefits of path switching, and demonstrated its feasibility with the help

TABLE III

VOICE QUALITY IMPROVEMENT WITH PATH SWITCHING: EXPERIMENTAL RESULTS

src	dst	Path 1	Path 2	Ideal gain	Actual gain	α (%)	Duration (min)
UMass	UMN	UMass-UPenn(AB)-UMN(CO)	UMass-UPenn(VE)-UMN(AB)	0.45	0.36	81	20
UMass	UPenn	UMass-UPenn(CO)	UMass-UMN(AB)-UPenn(GE)	0.31	0.26	83	165
UMN	UMass	UMN-UPenn(AB)-UMass(AB)	UMN-UPenn(AB)-UMass(CO)	0.15	0.10	69	78
UMN	UPenn	UMN-UPenn(GE)	UMN-UMass(AB)-UPenn(VE)	0.08	0.05	62	10
UPenn	UMass	UPenn-UMass(AB)	UPenn-UMass(CO)	0.19	0.14	73	21
UPenn	UMN	UPenn-UMN(AB)	UPenn-UMN(CO)	0.87	0.83	95	44

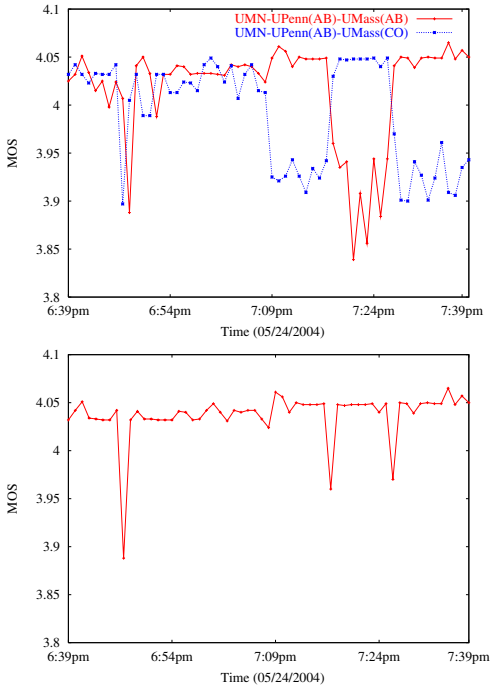


Fig. 11. A typical example that shows how path switching avoids quality degradations: (a) quality variations on the two paths (top), and (b) the resulting quality when path switching is applied (bottom).

of a prototype application-driven path switching gateway. We concluded that, with sufficient path diversity, i.e., paths with reasonably uncorrelated performance variations, path switching is indeed capable of yielding meaningful improvements in voice quality. Our investigation also established several other important conclusions, and in particular that accounting for network performance variations in terms of their impact on application quality was key to making judicious path switching decisions. Our experiments also highlighted the benefits of adaptive decisions, especially in light of the often changing nature of the time scale at which network congestion takes place. Our study suggests that by exploiting the inherent path diversity of the Internet, application-driven path switching is a viable option in providing quality-of-service to applications. We believe that with the deployment of increasingly disparate technologies (in particular, wireless networks), Internet paths will become ever more diverse. Intelligently creating and exploiting path diversity via mechanisms such as overlays and dynamic path switching is therefore an important avenue to meet and improve the quality-of-service of applications. We plan to pursue these issues further in the context of hybrid wired/wireless networks and other applications such as video.

VII. ACKNOWLEDGMENTS

This work is supported by the National Science Foundation under the grants ANI-9906855, ANI-0073819, ITR-0085930, ITR-0085824, and CNS-0435444.

We also would like to thank Peter Bartz, Frank Di Gravina, David Farmer, and Chris Hertel at UMN, Tyler Trafford at UMass, and the ISC department at UPenn for their support and assistance in setting up the testbed.

REFERENCES

[1] V. Cerf and R. Kahn, "A protocol for packet network interconnection," *IEEE Trans. Commun. Technol.*, vol. COM-22, no. 5, May 1974.

[2] S. Savage, A. Collins, E. Hoffman, J. Snell, and T. Anderson, "The end-to-end effects of Internet path selection," in *Proc. of ACM SIGCOMM*, September 1999.

[3] A. Akella, B. Maggs, S. Seshan, A. Shaikh, and R. Sitaraman, "A measurement-based analysis of multihoming," in *Proc. of ACM SIGCOMM*, August 2003.

[4] J. G. Apostolopoulos, T. Wang, W.-T. Tan, and S. Wee, "On multiple description streaming with content delivery networks," in *Proc. of IEEE INFOCOM*, June 2002.

[5] T. Nguyen and A. Zakhor, "Path diversity with forward error correction (PDF) system for packet switched networks," in *Proc. of IEEE INFOCOM*, April 2003.

[6] "RouteScience," http://www.routescience.com/.

[7] "Internap," http://www.internap.com/.

[8] D. G. Andersen, H. Balakrishnan, M. F. Kaashoek, and R. Morris, "Resilient overlay networks," in *Proc. of SOSP*, October 2001.

[9] D. G. Andersen, A. C. Snoeren, and H. Balakrishnan, "Best-path vs. multi-path overlay routing," in *Proc. of Internet Measurement Conference*, October 2003.

[10] S. Tao, K. Xu, Y. Xu, T. Fei, L. Gao, R. Guerin, J. Kurose, D. Towsley, and Z.-L. Zhang, "Exploring the performance benefits of end-to-end path switching," in *Proc. of ICNP*, October 2004.

[11] "Skype," http://www.skype.com/.

[12] "Methods for subjective determination of transmission quality," *ITU-T Recommendation P.800*, August 1996.

[13] "Perceptual evaluation of speech quality (PESQ): An objective method of end-to-end speech quality assessment of narrow-band telephone networks and speech codecs," *ITU-T Recommendation P.862*, January 2001.

[14] "The e-model, a computational model for use in transmission planning," *ITU-T Recommendation G.107*, March 2003.

[15] R. G. Cole and J. H. Rosenbluth, "Voice over IP performance monitoring," *Computer Communication Review*, vol. 31, no. 2, pp. 9–24, April 2001.

[16] C. Perkins, O. Hodson, and V. Hardman, "A survey of packet loss recovery techniques for streaming audio," *IEEE Network*, September/October 1998.

[17] L. Ding and R. A. Goubran, "Speech quality prediction in VoIP using the extended E-model," in *Proc. of IEEE GLOBECOM*, December 2003.

[18] A. Bremler-Barr, E. Cohen, K. Kaplan, and Y. Mansour, "Predicting and bypassing end-to-end Internet service degradations," *IEEE J. Select. Areas Commun.*, vol. 21, no. 6, August 2003.

[19] S. B. Moon, J. Kurose, and D. Towsley, "Packet audio playout delay adjustment: Performance bounds and algorithms," *ACM Multimedia Systems*, vol. 6, no. 1, pp. 17–28, January 1998.

[20] Luigi Rizzo, "Dummynet," http://info.iet.unipi.it/luigi/.