SureCall: Towards Glitch-Free Real-Time Audio/Video Conferencing

Amit Mondal
Northwestern University
Evanston, IL, USA
a-mondal@northwestern.edu

Ross Cutler, Cheng Huang, Jin Li
Microsoft Corporation
Redmond, WA, USA
{rcutler, chengh, jinl}@microsoft.com

Aleksandar Kuzmanovic
Northwestern University
Evanston, IL, USA
akuzma@cs.northwestern.edu

***Abstract*—Global enterprises are increasingly adopting *unified communication solutions* over traditional telephone systems. Such solutions provide integrated audio/video conferencing and messaging services, and enable flexible working environments by allowing mobile and dispersed users to communicate and collaborate easily and efficiently. The ultimate goal of unified communications is to ensure a smooth and best possible user experience across all scenarios.**

To address this challenge and understand the impact of various network scenarios on unified audio/video conferencing, we have developed a distributed experimental platform – SureCall – and deployed it on over 80 machines across a global enterprise and many residential networks. SureCall has collected worth of more than 6 months of packet-level audio/video conferencing traces. Through in-depth analysis of these traces, we have quantitatively compared how key performance metrics, such as packet loss and jitter, as well as the correlation between them, are affected by the enterprise and residential networks, by WiFi connections and VPN links, etc. In addition, we show how SureCall can serve as an ideal platform to design, experiment and validate new schemes and algorithms. We have developed a new audio quality classifier using the SureCall platform, which is being experimented with the recent release of Office Communicator solution for large-scale validation.

I. Introduction

IP based audio/video conferencing is expected to eventually replace traditional telephone systems (PBX and PSTN) for enterprises. Industry leaders like Cisco and Microsoft are spending billions of dollars to provide enterprise grade audio/video conferencing solutions [1], [2]. These solutions provide improved flexibility over traditional telephony systems at significantly reduced costs. More importantly, such solutions offer a *unified communication experience*, which enables mobile and dispersed users to communicate and collaborate easily and efficiently, irrespective of user location (on the enterprise campus, at home, or while traveling), device being used (computer or smart phone), and network connected to (enterprise, residential, wireless, or VPN). The ultimate goal of unified communications is to ensure a smooth and best possible user experience across all scenarios.

To understand user experience in unified communications, we set out to characterize the performance of real-time audio/video conferencing under various scenarios. We develop a distributed measurement platform, SureCall, to gather packet level traces of synthetically generated VoIP and video conferencing traffic; and analyze packet traces to quantitatively characterize the impact of various network scenarios on the performance of real-time audio/video conferencing.

There has been a substantial amount of research to understand the performance of VoIP and video communications over the Internet (*e*.g. [3]–[9]). Chen *et al.* [3] focus on the QoS of Skype VoIP system. Boutreams *et al.* [5] analyze VoIP performance on the Sprint network. Markopoulou *et al.* [4] compare the VoIP performance across a number of ISPs. In [9], the authors assess the call success probability, the call abortion probability induced by network outages, as well as the proportion of time that the network is suitable for VoIP service. In [8], the authors characterize the loss, delay and jitter of VoIP traffic using the traces collected from Internet backbone. Different from the above work, this study is based on a unique data collection of large-scale end-to-end packet-level traces from the SureCall platform. Furthermore, none of the above work compares and contrasts the difference for the same set of users across a wide variety of scenarios, which we characterize in this study.

The main contributions of the paper are threefold.

- We design, develop, and deploy SureCall to over 80 machines across a global enterprise and many residential networks, and collect worth of more than 6 months of packet-level traces of synthetically generated audio/video conferencing traffic (Section II).
- Analyzing the packet traces, we quantify the impact of various network scenarios on the performance of audio/video conferencing (Section III and IV). Our key findings are as follows: (i) jitter and loss in the residential networks are an order of magnitude higher than in the enterprise network; (ii) relative degradation in jitter and loss due to WiFi connections is significantly worse in the enterprise network than in the residential networks; (iii) VPN links can greatly increase jitter and loss; (iv) in both the enterprise and residential networks, end-to-end delay increases substantially before packet loss events. In the residential networks, higher delay increase also corresponds to longer loss burst. This, however, is *not* the case in the enterprise network.
- We show how SureCall can serve as an ideal platform to design, experiment and validate new schemes and algorithms (Section V). Using the SureCall traces, we have trained a new classifier that can accurately predict

when network issues are most likely to cause audio quality degradation. The classifier is being experimented with the recent release of Office Communicator solution for large-scale validation. Using the SureCall platform, we also propose, experiment and validate a *WiFi Relay* solution, which uses heavy application-level replication through relays to significantly improve VoIP performance for WiFi users.

II. SureCall platform

A. SureCall Architecture

The SureCall platform is comprised of a light-weight master controller, which serves as the central coordinator, and clients, which run on volunteers' machines. The master controller (master henceforth) maintains a persistent connection with each client, and keeps the latest status (online/offline, idle or active in conferencing) of the client.

The master schedules clients to emulate conferencing by sending instructions to start an audio/video conferencing session between them. The conference session can be audio only or audio together with video. Bitrate and frame structure (e.g., the size and frequency of audio frames, or those of video frames) are specified by the master.

Clients implement functionalities to emulate audio/video conferencing, which we briefly summarize here and elaborate further through the rest of this section:

- establishing UDP connections in both directions;
- sending emulated audio/video traffic, and recording trace details in compressed binary format;
- measuring network connectivity close to the clients and recording details;
- recording environmental details on client machines, such as CPU load and network interface type.

B. Implementation and Automatic Upgrade Mechanism

We develop SureCall for the `WINDOWS` platform using `C#` on `.NET` platform. An important design decision is to make SureCall clients upgradeable without user intervention. SureCall is designed in such a way that an upgrade is completely transparent to end users. We divide the functionalities of SureCall into two major components: a bare minimum framework and an upgradeable assembly (e.g., `DLL`). The framework runs as a `WINDOWS` service and starts as soon as a machine boots up. Its essential functionalities are monitoring the status of the assembly and initiating an upgrade when a new version becomes available. The assembly is loaded as a dynamic module. Once a new version is ready, the old one can be unloaded on-demand and the new assembly is loaded.

Whenever an upgrade is ready for deployment, there are two ways to trigger clients to download the new assembly. The master can notify the clients via the persistent connections. Alternatively, the clients can pull information about the new update from a pre-determined `URL` upon the reboot of volunteers' machines.

SureCall provides the volunteers with the flexibility to stop and restart the service at any point of time. SureCall creates a "system tray" icon with which the users can easily control the SureCall application.

C. Deployment of SureCall Client

	N. America	Europe	Asia	Oceania	S. America
# of city	58	23	13	3	2
# of IP	473	124	150	6	3

TABLE I
Endpoints Connected from Home

	N. America	Europe	Asia	Oceania	S. America
# of city	12	8	4	2	2
# of IP	1023	122	9	9	33

TABLE II
Endpoints within a Global Enterprise

We recruit volunteers from the Microsoft global enterprise to install SureCall on their workstations, laptops, as well as on their home machines. To create clean and separate *home*[1] and *enterprise* scenarios, we run two separate masters: one on the public Internet and the other within the enterprise network. A SureCall client first attempts to connect to the master within the enterprise network. It connects to the master on the public Internet only when the attempt to connect to the enterprise master fails.

The SureCall platform has been operating since September, 2008. It currently runs on 80 unique machines across five continents, out of which 32 connect only within the enterprise, another 20 connect only from home, and the remaining 28 move between enterprise and home from time to time. Between September, 2008 and January, 2009, using SureCall we have collected more than 4,800 hours worth of emulated audio/video conferencing traces (over 700 hours from home and over 4,100 hours from enterprise), which, in the rest of the paper, are referred to as *home trace* and *enterprise trace* respectively.

We observe 1,952 different IP addresses in the collected traces, out of which 1,196 are within enterprise and 756 are from home. The large number of IP addresses is due to DHCP used within the enterprise network, by the DSL and cable service providers, as well as by the volunteers connecting from different locations while traveling. Table I and II show the geographical locations of those IP addresses.

D. Audio and Video bit rates

In the current deployment of SureCall, a client participates in conferencing at most once per hour. Each audio/video session lasts five minutes. In an audio session, a 60-byte UDP packet is sent every 20 $msec$, at a bitrate of 24 $Kbps$. In a video session, there are three types of frames: I-frame, SP-frame, and P-frame. An I-frame comprises five back-to-back packets and is sent once every 10 $seconds$. A SP-frame carries three back-to-back packets and is sent once every $second$. A P-frame comprises a single video packet and is sent every 66

[1]We will use home and residential interchangeably in the paper.

$msec$. Each video packet is 1400 bytes, thus at an average bitrate of approximately 192 $Kbps$.

E. What data is collected?

Each audio/video packet carries the following information: (i) the timestamp from the sender machine, (ii) the packet sequence number, (iii) (in the case of video traffic) the packet type indicating a frame type (I/SP/P), and (iv) the time elapsed since the last packet is sent (`sndgap`).

For each received packet, the following information is logged in a compact binary trace: (i) the receiving timestamp from the receiver machine, (ii) the sending timestamp from the sender machine, (iii) the packet sequence number (iv) packet type (v) the `sndgap` information contained in the packet (vi) the time elapsed since the last packet is received (`rcvgap`), and (vii) the CPU load when the packet is received.

F. Handling NAT boxes

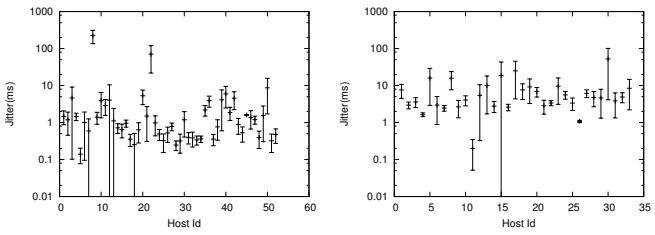
Most of the home machines in our deployment connect to the Internet via a home router. This creates two challenges: first, the master cannot actively establish connection to the clients, and second, clients cannot communicate with each other directly. The first challenge is solved by maintaining a persistent connection between a client and the master. Clients connect to the master as soon as SureCall starts.

To solve the second problem, we have implemented the STUN (Simple Traversal of User Datagram Protocol Through Network Address Translators) NAT traversal protocol [10]. A client uses a mediator which is universally accessible on the public Internet, to resolve the NAT box port number associated with the other endpoint's socket. We found that 64% of direct calls succeeded in the home deployment.

G. Limitations of SureCall

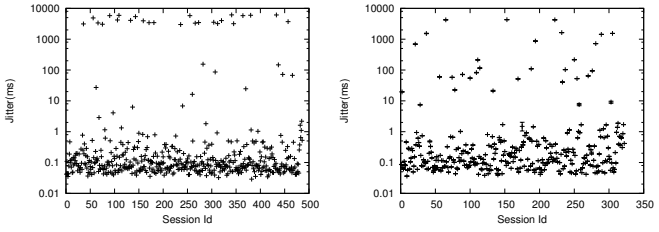
The current deployment of SureCall chooses a constant bit rate for both audio and video communications. However, most modern audio/video conferencing applications use sophisticated codec to change audio/video coding bit rate based on changing network conditions. We believe that it will be a very valuable future study to understand the interplay between network conditions and adaptive algorithms, which can be easily accommodated by SureCall's automatic upgrade capability.

Another limitation is that the scheduling of SureCall measurements does *not* take into account hidden factors that might also impact performance. For instance, when a SureCall session is scheduled, our volunteers could be downloading large software packages, or other family members sharing same residential networks might be running P2P sharing applications. These will inevitably affect our data collection and analysis. Nevertheless, such factors can also exist in real audio/video conferencing scenarios. Therefore, we do *not* regard that our data collection is contaminated by such factors.



(a) Enterprise (b) Home

Fig. 1. Jitter distribution across hosts.



(a) Host-I (b) Host-II

Fig. 2. Jitter distribution across sessions.

III. Data Processing

In this section, we will discuss trace preprocessing, trace classification, and the methodology we apply to detect hosts with persistently poor network connectivity. In trace preprocessing, we first compare widely referenced clock skew estimation algorithms, and then choose the right algorithm to compensate clock skew in SureCall traces.

A. Handling Clock Skew

We use *one-way transit time* (OTT), obtained by subtracting the sending time (in sender's clock) from the receiving time (in receiver's clock), to infer network condition changes. It is well known that clocks in different machines can run at different speeds, and may not be synchronized with the *true* time by national standard. The *relative* clock speed difference between the machines that are sending audio/video traffic to each other (referred to as *relative clock skew*), plays a significant role in the accuracy of the inference. If proper care is not taken, enlarging OTTs caused by relative clock skew will lead to false conclusions about worsening network conditions.

Due to the use of cheap crystals in modern computers, relative clock skew can be very significant. Figure 3 shows

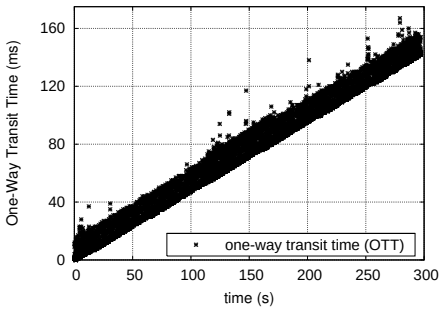


Fig. 3. Clock Skew in the Wild

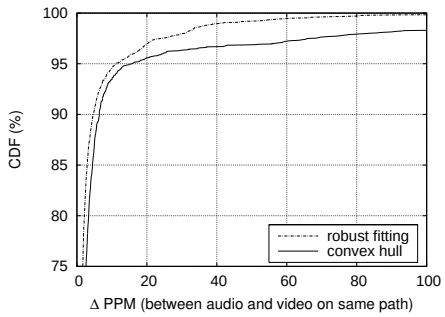


Fig. 4. Comparison of Skew Estimation Algorithms

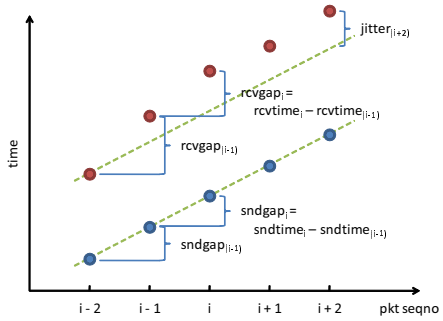


Fig. 5. Jitter calculation

the one-way transit time in one of the traces in the collected dataset. The relative clock skew in this example is more than $150\,ms$ within 5 minutes.

1) Clock Skew Estimation Algorithms: Several algorithms have been proposed to estimate relative clock skew between two machines. We compare two widely referenced algorithms. (i) The first algorithm was proposed by Paxson *et al.* back in 1998 [11]. It first segments the OTT measurements into a number of buckets each with a certain time span based on the arrival timestamps of these OTTs. It then selects the minimum OTT in each bucket to form a *de-noised* OTT sequence. Next, it applies robust fitting techniques to find a linear slope (*e.g.*, calculating all pair-wise slopes and picking the median), which best represents the trend of the de-noised OTT sequence. We refer to this method as *robust fitting*. (ii) The second algorithm was proposed by Moon *et al.* [12], and was further explored by Zhang *et al.* [13]. Conceptually, it attempts to find a linear slope such that all OTTs stay above the line, and the total deviations of the OTTs from the line are kept at a minimum in terms of certain metrics. In [12], the vertical distance between each OTT and the line is chosen as the metric. We refer to this method as *convex hull* since the estimated clock skew slope always aligns with the convex hull formed by all the OTTs.

2) Comparing and Choosing the Right Algorithm: We use SureCall traces to make a realistic comparison of these two clock skew estimation algorithms. In the absence of the ground truth, we resort to compare these algorithms in a relative sense. In particular, we use data collected when there were concurrent audio and video conferencing sessions between the same pair of machines. We estimate relative clock skews independently from audio and video traffic. Ideally, these two estimations should yield similar results. Thus, examination of the relative difference can shed light on the accuracy and robustness of different algorithms.

We calculate the difference between the estimated clock skews from each concurrent audio and video conferencing trace, and plot the cumulative distribution from all such traces. Figure 4 shows that the difference is much larger using the convex hull method than using robust fitting or linear regression. In fact, with convex hull approach 2.5% of the samples have values of 60 PPM (parts per million) or more, which is equivalent to $18\,ms$ in a 5-minute period, which is quite significant. We conclude that the robust fitting method is more robust than the convex hull method. Indeed, detailed examination of the traces reveals that the requirement that all the OTTs stay above the estimated clock skew slope is too strong. Merely one outlier with an abnormally low OTT, which is not absent in our trace collection and could happen due to random anomaly in machine clocks, can significantly affect the clock skew estimation. We apply the robust fitting algorithm to compensate clock skew in our dataset.

B. Jitter Computation

Jitter is defined as the time difference between an actual packet receiving time and an ideal receiving time. As illustrated in Figure 5, packets are transmitted by a sender at certain times (denote the send time of packet i as $\texttt{sndtime}_i$). Assuming the network condition is perfect (e.g., congestion-free, so no transmission time variation), then these packets will arrive at a receiver after a fixed time offset (denote such ideal receiving time of packet i as $\texttt{rcvtime}_i^0$). However, as network conditions vary (which is norm), the arrival times are affected and, as a result, they deviate from ideal receiving times (denote the actual receiving time of packet i as $\texttt{rcvtime}_i$). Therefore, we define jitter of packet i as $\texttt{jitter}_i = \texttt{rcvtime}_i - \texttt{rcvtime}_i^0$. It can be shown that $\texttt{jitter}_i = \texttt{rcvgap}_i - \texttt{sndgap}_i$ if $\texttt{jitter}_{i-1}$ is zero. Thus, we use `sndgap` and `rcvgap` information to compute packet jitters.

C. Trace classification and Stratification

We classify the traces into *intra-continental* (*US-US* for those within United States too) and *inter-continental* based on the locations of the endpoints. We use the Quova Geolocation [14] database to find the geographical locations of the external endpoints and Microsoft's Internal Geo-Database to locate enterprise endpoints.

We also classify the traces based on the type of network connectivity used by endpoints during conferencing sessions. If both endpoints use wired connection then we classify the associated trace into *wired* category. Similarly, if at least one of the endpoints uses wireless connection then we classify the associated trace into *wireless* category. If any of the endpoints uses VPN then we classify the associated trace into *VPN* category. Finally, we classify audio traces into *audio-only* category or *audio+video* category, based on whether there are simultaneous video sessions.

These classifications stratify the measurements to account for possible confounding factors, and make it possible to systematically study the impact of each individual factor.

D. Sanity Check of Trace Collection

In case some volunteer machines are faulty, which can constantly contribute abnormal measurements, we conduct the following sanity check on our data collection. We compute median jitter values for all intra-continental audio sessions and aggregate them based on the receiver identifier. Figure 1(a) and (b) show the average of the median jitter values with a 95% confidence interval for hosts that appear in the enterprise and home traces.

Figure 1(a) shows that most enterprise hosts have small jitter characteristics with the exception of two hosts, which have orders of magnitude larger jitter value. We then plot the individual median jitter values of the sessions involving these two hosts in Figure 2(a) and (b). Figure 2(a) and (b) show that most sessions experience very little jitter, except for a very few sessions which observe few orders of magnitude higher jitter. This might be due to occasional poor network connectivity or highly network intensive applications running at those endpoints. A similar calculation for loss rate shows that none of the endpoints consistently exhibits a very high loss rate. Therefore, we conclude that there is *no* faulty endpoint and the entire data collection is useful.

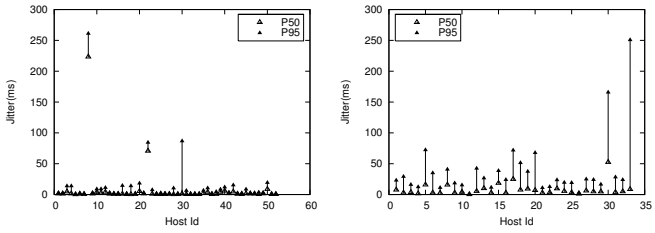
IV. Analyzing SureCall Traces

In this section, we analyze SureCall traces to quantify the impact of various network components on audio/video conferencing quality.

A. Enterprise vs. Residential Networks

Unified communication solutions are expected to offer smooth user experience across enterprise and residential networks. To understand the challenge, we conduct a comparative study between the two types of networks. Using the traces collected from the SureCall platform and through jitter and packet loss analysis, we draw quantitative conclusions on how enterprise and residential networks impact the quality of audio/video conferencing.

1) Jitter: We quantify the performance gap between enterprise and home networks in terms of network jitter as observed by the SureCall traces. We compute the 50^{th} and the 95^{th} percentile of the jitter values for each US-US wired audio-only session, and plot the corresponding distributions in Figure 7. The figure shows that residential networks have significantly worse jitter characteristics compared to enterprise networks. In addition, we plot the 50^{th} and the 95^{th} percentile of the jitter values for each endpoint in Figure 6. Although the median jitter value in both residential and enterprise networks is low, the 95^{th} percentile jitter value gives a different picture. We observe that the jitter variation is much higher in residential networks than in enterprise networks, which will lead to more observable degradation of audio and video conferencing experience in residential networks.



(a) Enterprise (b) Home

Fig. 6. The 50^{th} and 95^{th} percentile of jitter distribution across hosts. Jitter variation is much higher in residential networks than in enterprise networks. Figure 6(b) shows that 95^{th} percentile jitter values is significantly worse than the median jitter values in home networks.

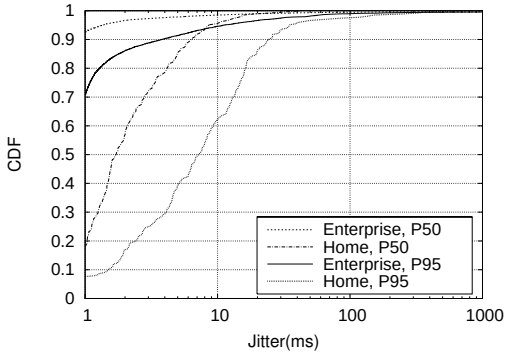


Fig. 7. Jitter Distribution (US-US, wired traces).

When endpoints outside of US are considered, the performance gap between residential networks and enterprise networks becomes even wider. Figure 8 shows the distributions of the 50^{th} and the 95^{th} percentile of the jitter values in the *inter-continental wired audio-only traces*. Compared to Figure 7, we can see the jitter in the residential network significantly increases over long distance compared with the enterprise network. In particular, in the inter-continental traces, for more than 10% of the sessions, the 95^{th} percentile of the jitter values is more than $100\,ms$, which is a typical upper bound of the *de-jitter* buffer size in audio/video conferencing applications. Thus, jitter can cause significant quality degradation in the inter-continental audio/video conferencing scenario.

2) Packet Loss: We compare the packet loss behavior of residential and enterprise networks, in both short and long-time scale. In particular, we use detailed SureCall trace to study

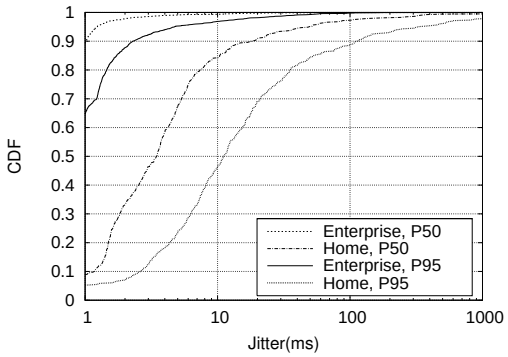


Fig. 8. Jitter Distribution (inter-continental, wired traces).

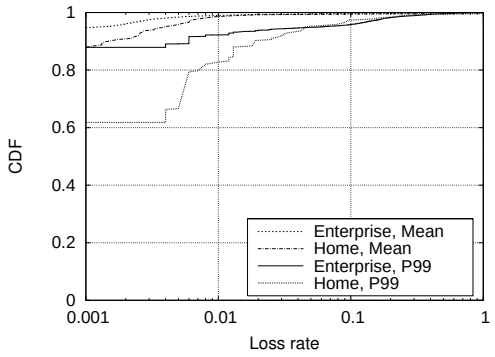


Fig. 9. Loss Rate Distribution (US-US, wired traces).

whether packet loss is random or bursty in the real world. While Forward Error Correction (FEC) techniques and loss concealment techniques [15] can be used to recover/conceal random or small-size burst losses, large bursty packet losses are known to cause severe quality degradation in audio/video conferencing [16].

To analyze short-term loss, we slice each audio session into 5-second segments, and compute the average loss rate during each 5-second segment. We then compute the 99^{th} percentile of the loss rate values for each session and obtain a distribution of these 99^{th} percentile values. For long-term loss, we compute the average loss rate during the entire duration for each session. Figure 9 compares the mean and the 99^{th} percentile of loss rate in residential and enterprise US-US wired audio-only traces. It is surprising that more than 5% of enterprise sessions experience periods (or 5-second segments) with a loss rate greater than 10%. This suggests that even well provisioned enterprise networks can have bad network behavior in short time scale.

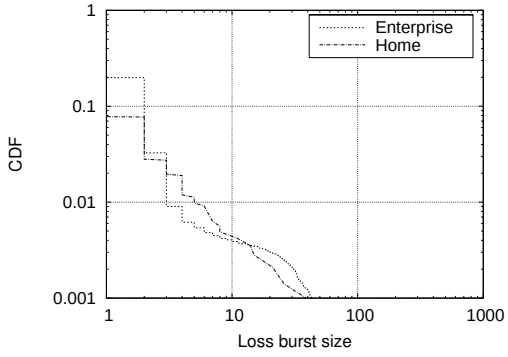


Fig. 10. Loss Burst Size Distribution (US-US, wired traces).

We also calculate the burst size distribution in both the residential and enterprise traces. We count the number of consecutive packet losses during all loss events. Figure 10 shows the CCDF (complementary cumulative distribution function) of loss burst size for both the home and enterprise traces. Though 80% of loss events in enterprise and 92% of loss events in home are only a single packet long, both enterprise and home networks show a long tail in the loss burst size distributions. It is a non-trivial percentage of loss events where more than 10 consecutive packets are lost.

B. WiFi Connections

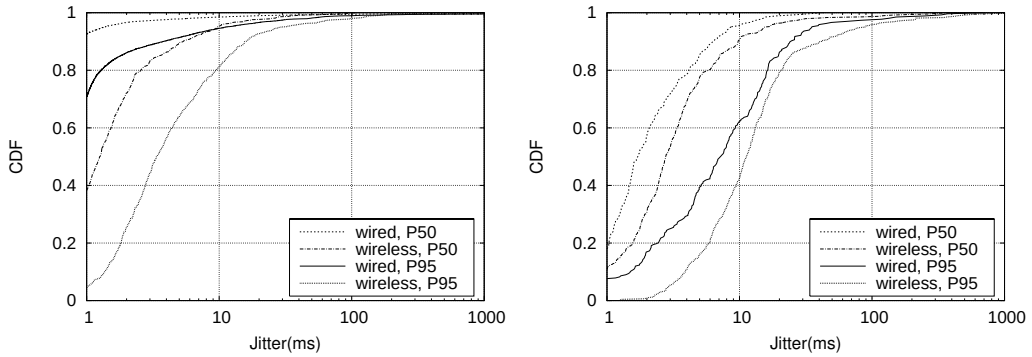
An increasing number of users are connecting to networks through WiFi, both in enterprise and at home. A recent survey [17] shows that 43% of small businesses provide only WiFi connections to their employees and 36% of organizations use VoIP over WiFi. Over WiFi links, packet loss is more likely to happen due to bad connectivity, weak WiFi signal, and high interference, etc. [18]. A recent large-scale study [19] shows that VoIP sessions over WLAN experience significant quality degradation even in a well provisioned enterprise network. In this subsection, we analyze the SureCall traces and quantitatively study the performance degradation caused by WiFi links.

We classify the *US-US audio-only traces* based on the network interfaces used by endpoints during conferencing sessions and compare jitter and loss characteristics between the *wired* and *wireless* traces. Figure 11 shows the impact of WiFi links on jitter in both enterprise and home networks. Figure 12 quantifies the loss rate degradation due to WiFi links. In both enterprise and home networks, wireless traces have significantly worse jitter and loss statistics than the wired traces. More than 10% of the enterprise wireless sessions experience a medium loss rate of more than 1%. Around 10% of the sessions even experience periods with a loss rate of more than 10%. It is interesting to see that the degradation due to WiFi links in the enterprise scenario is more severe than that in the home scenario. This might be explained by dense WiFi access point deployments in enterprise, and a higher number of users competing for wireless channels.

C. VPN Links

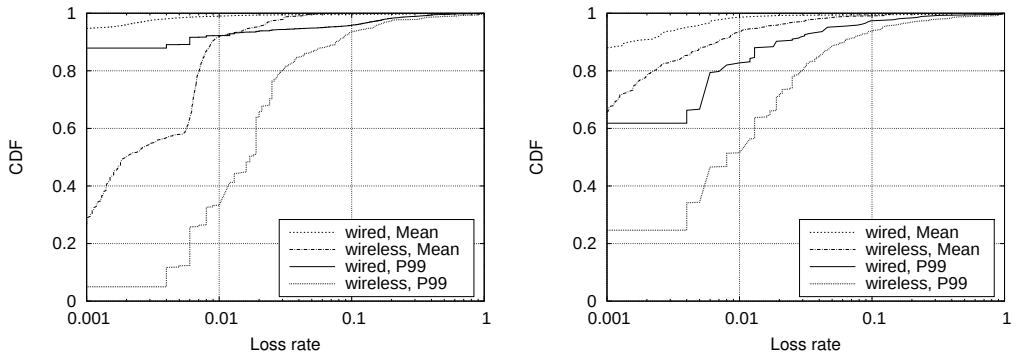
Many telecommuting users connect to enterprise networks through Virtual Private Networks (VPN), where VPN packets are tunneled using Point to Point Tunneling Protocol (PPTP), Layer 2 Tunneling Protocol over Internet Protocol Security (L2TP/IPSec), and Secure Socket Layer (SSL). All packets entering and exiting enterprise networks pass through VPN servers. Such VPN servers can sometimes get overloaded, causing performance degradation.

We isolate *US-US audio-only enterprise traces* where one endpoint is connected to enterprise networks from outside using VPN connections and the other endpoint is located inside enterprise networks using a wired connection. We have 80 hours worth of traces in this category. We then compare the jitter and loss statistics of these VPN traces with *US-US audio-only wired enterprise traces*. Figure 13 shows the impact of VPN connections on jitter and loss characteristics. For more than 5% of the VPN sessions, the 95^{th} percentile of the jitter values is more than $100\,ms$. VPN connections also worsen the loss characteristics, *e*.g. more than 20% of the sessions experience periods with a loss rate greater than 10%[2].



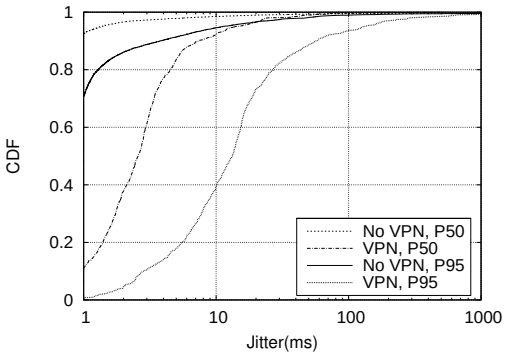
(a) Enterprise (b) Home

Fig. 11. Impact of WiFi Connections on Jitter.

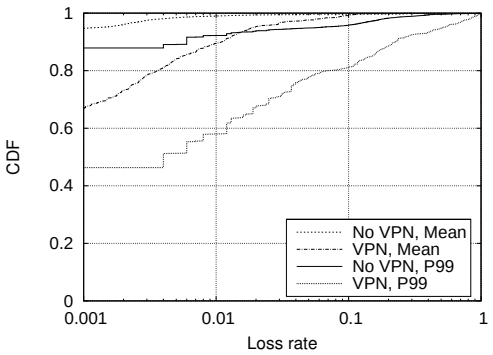


(a) Enterprise (b) Home

Fig. 12. Impact of WiFi Connections on Packet Loss.

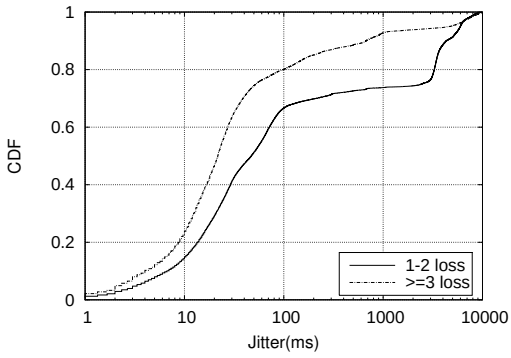


(a) Jitter

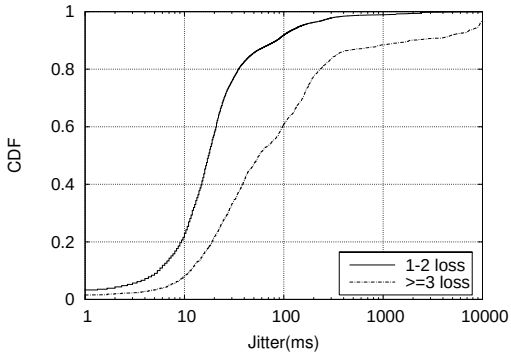


(b) Packet Loss

Fig. 13. Impact of VPN Links.



(a) Enterprise



(b) Home

Fig. 14. The correlation between loss and jitter. End-to-end delay increases significantly before loss events in both enterprise and home networks.

D. Correlation between Jitter and Packet Loss

We study the extent to which packet loss and jitter are correlated, i.e. whether abrupt jitter increase can serve as a precursor of network congestion and predict future loss events so that audio/video conferencing applications can take anticipatory action. Moon et al. [20] conducted an early study to understand packet delay and loss correlation in the Internet. Based on traces collected from a few Internet paths, they show that packets sent closely before loss events indeed experience high delay. From our large-scale SureCall traces, we first conduct similar analysis and reach the same conclusion. Furthermore, we analyze the correlation between jitter and loss burst size. Surprisingly, we observe complete opposite trends for enterprise and home networks.

We calculate the average increase in end-to-end delay for the last three packets preceding a loss event. We found that more than 82% of the time in enterprise networks, and around 80% of the time in home networks, there is an increase of at least 10 ms in the end-to-end delay before a loss event. Therefore, the increase in end-to-end delay can indeed be used as a precursor of a loss event. In addition, we compute loss burst size, which is the number of consecutive packet losses during the loss event. Figure 14 shows the CDF of the increase in end-to-end delay for different loss burst size in the *US-US audio-only wired traces*. Figure 14(b) indicates that the average increase in end-to-end delay grows with longer bursts of packet losses in home networks. Surprisingly, Figure 14(a) shows that enterprise networks behave quite differently. In particular, the CDF curve corresponding to longer bursts (three or more consecutive losses) shifts to the left of that corresponding to single or double packet losses. This suggests that there are severe packet losses without a precursor rising delay. Our interpretation is that in enterprise network, the link bandwidth is high and the end-to-end propagation delay is low, which leads to more bursty traffic for TCP. Thus, the congestion event in enterprise network happens more abruptly.

V. Applications of SureCall

In the previous section, we use SureCall to quantify the impact of the quality of audio/video conferencing under various network scenarios. Based on the understanding of network behavior unveiled by SureCall, new audio/video conferencing algorithms can be designed. In this section, we report initial studies along this direction, where SureCall is served to rapid prototype and validate schemes and algorithms before they are pushing into real production systems.

A. Network Audio Diagnostics

Most VoIP systems include audio concealment methods to try to compensate for network impairments such as packet loss or jitter. Lost audio packets can be recovered through the use of forward error correction (FEC). Moreover, unrecovered lost packet may be further concealed by interpolating or extrapolating the audio signals using models of speech signals. Jitter can be concealed through the use of audio de-jitter buffer. Modern audio decoder can even stretch or compress decoded audio so that the size of audio de-jitter buffer, which determines VoIP communication delay, can be adaptively adjusted so that it stays at a low level. As a result, network glitches, such as packet loss and jitter may not lead to an actual perceived audio glitch. In this section, we have trained a classifier through SureCall that can accurately predict when network issues will cause user perceived audio glitches. Our classifier considers the audio concealment algorithm incorporated in the de-jitter buffer and FEC. It uses two key statistics:

- concealed: percent of packets interpolated or extrapolated due to unrecovered packet loss after FEC
- stretched: percent of packets stretched via time compression

Our classifier operates as follows:

$$bad(trace) = \begin{cases} 1 & concealed > T_1 \text{ or } stretched > T_2 \\ 0 & otherwise \end{cases}$$

where T_1 and T_2 are arbitrary thresholds, and an output of 1 indicates that the network packet loss or jitter will lead to user perceived audio glitches. To train this classifier using supervised training methods we need to know if a given network trace will cause perceptible audio issues; that is, we need ground-truth data. This can be done objectively using a Perceptual Evaluation of Speech Quality (PESQ) tool that gives a measure of audio quality that is highly correlated with Mean Opinion Scores (see ITU-T P.862). In this standard, PESQ score less than 3 denotes unacceptable audio quality. The ground-truth for each network trace is determined by:

$$bad(trace) = \begin{cases} 1 & PESQ < 3 \\ 0 & otherwise \end{cases}$$

The following SureCall network traces are used:

- 108 Enterprise US-US Wired-Wireless traces
- 107 Enterprise US-US Wired-Wired traces
- 94 Home US-US Wired-Wireless traces

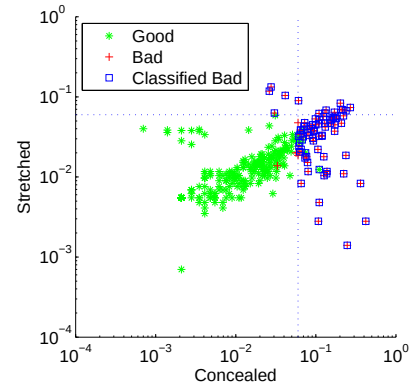
These traces were quasi-randomly selected to span PESQ score range. They are not uniformly sampled but are heavily weighted with bad calls; this gives more samples to estimate the true positive rate (TPR).

The results of this analysis are shown in Figure 15. Figure 15(a) plots the concealed and stretched statistics with ground-truth (Good or Bad) and classification results (Classified bad) using $T_1 = T_2 = 0.05$. Figure 15(b) is a Receiver Operating Characteristics (ROC) plot generated by varying $T_1 = T_2$ from 0.02 to 0.14. This classifier achieves a true positive rate of $> 80\%$ and a false positive rate of $< 1\%$ for a particular audio CODEC, RTAudio 16k [21]; results for other CODECs are similar. This new classifier is being experimented with the recent release of Office Communicator solution (release 14) for large-scale validation.

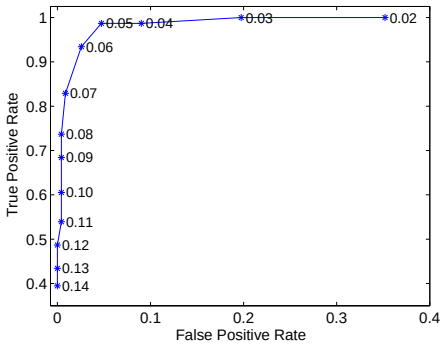
B. WiFi Relay

In the earlier section, we quantitatively analyze the network behavior of WiFi connections. The key conclusion is that, in

[2]Since one endpoint is located in home networks, it is possible that the effect of VPN connections is compounded with that of home networks.



(a) Enterprise



(b) Home

Fig. 15. Classifier Results Using RTAudio 16k.

both enterprise and residential networks, wireless connections incur significantly more packet losses than landlines. The detailed analysis in a companion study [22], however, shows that these losses can be effectively concealed by sending each packet up to five times, which we denote as *heavy replication*. Due to WiFi's inherent overhead, heavy replication only marginally increases WiFi airtime. Therefore, we propose a *WiFi Relay solution* to significantly improve the quality of audio conferencing through WiFi connections. In the solution, heavy replication only occurs between wireless endpoints and nearby wired relays, which is removed before packets are transmitted on inter-branch long haul links or the public Internet, to avoid the overhead on wirelines. Using the SureCall platform, we have implemented and experimentally validated the WiFi Relay solution. The results confirm that the solution can indeed greatly improve the performance of VoIP for WiFi users. We refer interested readers to the companion paper for details [22].

VI. CONCLUSION

In this paper, we present SureCall, a distributed experimental platform, to address the challenges of unified communications. Through large-scale traces collected in a global enterprise and many residential networks, we characterize the performance of real-time audio/video conferencing over a wide variety of network scenarios. Using the SureCall traces, we train a new classifier that can accurately predict when network issues are most likely to cause audio quality degradation. In addition, we report initial studies along the direction of improvements, and show how SureCall can serve as an ideal platform to experiment and validate new schemes and algorithms. Our future work includes analyzing and designing new FEC schemes to cope with burst loss patterns observed. Furthermore, while the primary subject of this paper is audio traffic, we are now focusing on video conferencing traffic and how to improve its quality and experience. In short, SureCall is a ripe platform and now serving as an important tool towards ultimate glitch-free audio/video conferencing.

REFERENCES

[1] "Cisco Unified Communications." [Online]. Available: http://www.cisco.com/en/US/netsol/ns151/networking_solutions_unified_communications_home.html
[2] "Microsoft Office Communicator." [Online]. Available: http://office.microsoft.com/en-us/communicator/default.aspx
[3] K. Chen, C. Huang, P. Huang, and C. Lei, "Quantifying skype user satisfaction," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 36, no. 4, Oct. 2006.
[4] A. Markopoulou, F. Tobagi, and M. Karam, "Assessment of VoIP quality over Internet backbones," in *IEEE INFOCOM*, Jun. 2002.
[5] C. Boutremans, G. Iannaccone, and C. Diot, "Impact of link failures on VoIP performance," in *NOSSDAV*, May 2002.
[6] I. Marsh, F. Li, and G. Karlsson, "Wide area measurements of voice over IP quality," *Lecture notes in computer science*, pp. 93–101, 2003.
[7] P. Calyam, M. Sridharan, W. Mandrawa, and P. Schopis, "Performance measurement and analysis of H. 323 Traffic," *Lecture Notes in Computer Science*, pp. 137–146, 2004.
[8] R. Birke, M. Mellia, M. Petracca, and D. Rossi, "Understanding VoIP from backbone measurements," in *IEEE INFOCOM 2007*, May 2007.
[9] W. Jiang and H. Schulzrinne, "Assessment of voip service availability in the current internet," in *PAM*, Apr. 2003.
[10] J. Rosenberg, J. Weinberger, C. Huitema, and R. Mahy, "STUN-simple traversal of user datagram protocol (UDP) through network address translators (NATs)," RFC 3489, IETF, Mar. 2003, Tech. Rep.
[11] V. Paxson, "On calibrating measurements of packet transit times," in *ACM SIGMETRICS Performance Evaluation Review*, Jun. 1998.
[12] S. Moon, P. Skelly, and D. Towsley, "Estimation and removal of clock skew from network delay measurements," in *IEEE INFOCOM*, Mar. 1999.
[13] L. Zhang, Z. Liu, H. Xia, C. Center, and Y. Heights, "Clock synchronization algorithms for network measurements," in *IEEE INFOCOM*, Jun. 2002.
[14] "Quova." [Online]. Available: http://www.quova.com/
[15] C. Perkins, O. Hodson, and V. Hardman, "A survey of packet loss recovery techniques for streaming audio," *IEEE Network*, vol. 12, no. 5, 1998.
[16] D. Hands and M. Wilkins, "A study of the impact of network loss and burst size on video streaming quality and acceptability," *Lecture notes in computer science*, pp. 45–58, 1999.
[17] C. MacGillivray, "Mobile Internet access in the SMB: Demand-side analaysis of WiFi and 3G mobile broadband," IDC, Dec. 2008.
[18] S. Rayanchu, A. Mishra, D. Agrawal, S. Saha, and S. Banerjee, "Diagnosing wireless packet losses in 802.11: Separating collision from weak signal," in *IEEE INFOCOM*, 2008.
[19] V. Vasudevan, S. Sengupta, and J. Li, "A first look at media conferencing traffic in the global enterprise," in *Passive and Active Measurement Conference*, Apr. 2009.
[20] S. Moon, J. Kurose, P. Skelly, and D. Towsley, "Correlation of packet delay and loss in the Internet," 1998, Tech Report, Univ. of Massachusetts, Amherst.
[21] "Overview of the Microsoft RTAudio Speech Codec," Microsoft Inc., 2006.
[22] A. Mondal, C. Huang, J. Li, M. Jain, and A. Kuzmanovic, "A Case for WiFi Relay: Improving VoIP Quality for WiFi Users," *IEEE ICC*, May 2010.