VIA: Improving Internet Telephony Call Quality Using Predictive Relay Selection

Junchen Jiang[1,2], Rajdeep Das[1], Ganesh Ananthanarayanan[1], Philip A. Chou[1], Venkata N. Padmanabhan[1]
Vyas Sekar[2], Esbjorn Dominique[1], Marcin Goliszewski[1], Dalibor Kukoleca[1], Renat Vafin[1], Hui Zhang[2,3]

[1] *Microsoft*, [2] *CMU*, [3] *Conviva*

ABSTRACT

The use of the public Internet for voice calls is here to stay. In spite of the volume and importance of Internet telephony, we have little quantitative understanding of (1) how network performance impacts user-perceived voice call quality, and (2) why and where such quality problems occur in the wild. To bridge this gap, we analyze a data set of 430 million calls from Skype, with clients spread across 1900 ASes and 126 countries. One of the key findings is that calls over bad networks are spread out geographically and over time.

To alleviate call quality problems, we present an architecture called VIA that revisits the use of classical overlay techniques to relay calls. We argue that this approach is both timely and pragmatic given that the private backbones built up in the recent years to connect globally distributed datacenters provide a readily available infrastructure for a *managed* overlay network. Trace-driven analysis shows that VIA can *potentially* improve up to 53% of calls whose quality is impacted by poor performance on key network metrics. An important research question, however, is whether these benefits can actually be realized in practice, in the face of significant spatial and temporal variability in performance and the existence of a large number of relaying choices. We develop a practical relay selection approach that intelligently combines prediction-based filtering with an online exploration-exploitation strategy. We demonstrate using both a large-scale trace-driven analysis and a small-scale real-world pilot deployment that VIA helps cut the incidence of poor network conditions for calls by up to 45% (and for some countries and ASes by even over 80%) while staying within a budget for relaying traffic through the managed infrastructure.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

SIGCOMM '16 Aug 22–26, 2013, Florianópolis, Brazil
© 2016 ACM. ISBN 978-1-4503-2138-9.
DOI: 10.1145/1235

CCS Concepts

•**Applied computing** → *Internet telephony;* •**Networks** → *Overlay and other logical network structures; Network performance analysis;* Network measurement;

Keywords

Internet telephony quality; Predictive relay selection; Managed overlay networks

1 Introduction

Over the last several years, we have seen a dramatic rise in Internet-based telephony, especially for long-distance international calling [5]. The importance of audio calling is evident from almost all major content and social networking platforms today offering some form of Internet calling capability (e.g., Skype, Google Hangouts, Facebook Messenger, WhatsApp, WeChat, FaceTime).

The key difference between Internet-based audio call streaming and on-demand video streaming such as Netflix is *interactivity*. The real-time interactive nature of audio calls make them far more sensitive to issues such as RTT, packet loss and jitter induced by the network [14] while at the same time making solutions based on application-level buffering, that work well with video streaming, less effective. Yet, despite the growing importance of Internet telephony and the dominant role that the network plays in user-perceived call quality, there have been few systematic studies *at scale* that analyze (1) how network performance impacts user-perceived quality of experience in the wild, and (2) the typical characteristics of performance issues in the real world.

As a first step to bridging this disconnect, we analyze measured network performance and user-perceived quality indexes from one of the largest deployed VoIP service, Skype, which serves hundreds of millions of users and handles a total call volume in excess of a billion talk-time minutes per day. Our dataset from Skype consists of a sample of this total call volume and includes 430 million call records from seven months spanning over 135 million users located across 126 countries. As expected, we observe that the call quality experienced by users is strongly correlated with the underlying network performance (RTT, packet loss rate and jitter). But contrary to our intuition, calls over bad networks are not confined to a few pockets but rather are spread out geographically and over time.

How might we alleviate these sources of poor network performance? Despite many prior efforts on network QoS, the Internet remains a best-effort network that provides no guarantees of performance. Also, the popularly used technique of deploying "edge" proxies close to users for better network performance does not cater to the client-to-client communication pattern of audio calls. If the caller and callee are in geographically distant locations, the VoIP application has no choice but to communicate over the wide-area network, which makes it challenging to meet the performance requirements of the application.

In this context, we argue that it is timely to revisit the idea of classical overlay routing (e.g., [7, 32]), which has seen well over a decade's worth of research. Overlay networks improve performance (though not guaranteed) by crafting paths that route around bottlenecks in the default Internet paths (e.g., BGP-derived). The adoption of overlay routing at scale has been stymied for various reasons but perhaps the most important one is the need to build up the overlay infrastructure (relay nodes) from scratch. Today, however, large cloud providers with geographically distributed datacenters that are connected by a "managed" backbone network provide the opportunity to construct highly performant overlay alternatives to the default Internet path [24, 36].

Inspired by this opportunity, we envision a framework called VIA, which can be viewed as an instance of a *managed overlay*. VoIP calls can be *selectively* relayed over managed overlay servers deployed by service providers like Skype. Microsoft could choose to route only (a subset of) Skype traffic through their managed network, and Google could do likewise with Hangouts traffic. Indeed, Skype has moved to a hybrid model where calls can be either relayed through datacenters or through direct peer-to-peer path, and Hangout has started to use multi-hop relays in the cloud for many calls in addition to the default Internet paths.

Managed overlays offer a pragmatic alternative to classical "user-defined" overlays as the providers can carefully provision the servers, decide which calls to relay, and control how the traffic is routed over the overlay, thus ensuring that the managed backbone network does *not* get turned into a general-purpose conduit. Also, by tying the use of the managed overlay to a widely-used application, traffic from the application itself could be used to learn about network conditions passively, thereby obviating the need for active probing whose overhead can be prohibitive at large scales.

We evaluate the potential of such relaying using trace-driven analysis of the Skype dataset using their production relays. We report that an "oracle" that looks into the future network performance and identifies the best relay(s) can improve 53% of calls whose quality is impacted by poor network performance (we derive thresholds for "poor network performance" in §2.2.) We believe this is *one of the largest potential analyses of overlay routing in production*.

Achieving this potential in practice, however, turns out to be challenging. The significant spatial and temporal variability in the network performance through the various relaying alternatives requires carefully tracking these dynamics. Also, operators may prefer to not overload their backbone network, thus requiring our solution to function within a "budget" for how much traffic could be relayed.

VIA uses a relay selection approach, called *prediction-guided exploration* to decide which calls to relay through the managed overlay and to pick the relay(s). It makes these decisions with only performance information from call history, which tends to be limited and highly skewed. The key insight behind our approach is the empirical observation that *even though available performance information from call history may not suffice to accurately predict the best relay for each call, it can nevertheless help identify a small subset of relay choices that contains the best relay*.

We use performance information from call history to filter out all but the most promising (*top-k*) relaying alternatives, from which, we then employ an online exploration-exploitation strategy to identify the optimal relay path, while staying within the relaying budget. Prediction-guided exploration, thus, strikes a balance between exploration-based approaches which explore all possible choices, and prediction-based approaches which attempt to predict a single best choice from history information. Note that prediction-guided exploration's objective is different than the classic machine learning problem of finding the *true top-k options* [10]. In addition, VIA also uses network tomography to expand the coverage of prediction to paths that have not even been seen.

Trace-driven simulation shows that VIA's improvement in call quality closely matches that of an oracle: VIA helps cut the incidence of poor network conditions for calls by up to 45% (and for some countries and ASes by even over 80%) We also implement a prototype of VIA with a cloud-based controller and modified Skype clients, and deploy and evaluate on a small testbed of 18 client pairs across five countries for relaying calls through Skype's managed relays.

Contributions: This paper makes three key contributions.

1. Analyzing, at scale, the impact of network performance on audio call quality. (§2)
2. Quantifying the potential benefits of a managed overlay network for improving audio call quality. (§3)
3. Highlighting the challenges in achieving these benefits and presenting a practical relay selection algorithm that delivers close-to-optimal performance. (§4)

2 VoIP Performance in the Wild

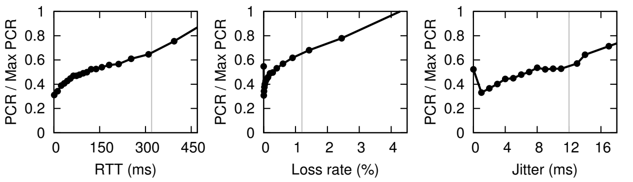
We use call logs from Skype (described in §2.1) to quantify the impact of network metrics on audio call quality (§2.2), and the patterns of poor network performance (§2.3 and §2.4). These observations motivate the need for and the design requirements of VIA.

2.1 Dataset description

The dataset from Skype consists of a sampled set of 430 million audio calls drawn from a seven month period. The sampled set includes both calls that use the default path (e.g., BGP-derived) between the caller and the callee as well as

Time	2015.11.15–2016.05.30
Calls	430M
Users	135M
ASes	1.9K
Countries/regions	126

Table 1: Dataset summary.



(a) PCR vs. RTT (b) PCR vs. Loss (c) PCR vs. Jitter

Figure 1: Individual network performance metrics have considerable impact on user experience (PCR); y-axis normalized to the maximum PCR value. Vertical gray lines show the thresholds for poor network performance.

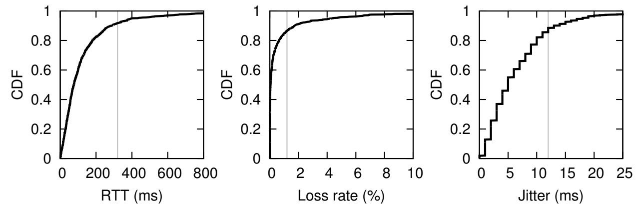
calls that are relayed through managed relay nodes distributed across datacenters in different locations. Note that today such relaying is typically employed for connectivity (e.g., firewall or NAT traversal) rather than for performance optimization (which is our focus here). Therefore, the only instances of relaying in our passively collected dataset correspond to the caller and callee being unable to establish a direct connection. Nevertheless, despite the bias, the dataset offers a *panoramic* view across a diverse set of end-points from over $1,905$ ASes across 126 countries. Table 1 summarizes its basic statistics.

To the best of our knowledge, our work is the first to study the quality of Internet telephony calls at such a large scale. There are other several characteristics that make this dataset stand out: most calls are international (46.6%), inter-AS (80.7%), and wireless (83%).These characteristics of the calls in the dataset allow us to study the performance of Internet telephony over a much greater diversity of Internet paths than has been considered in prior studies, where traffic was mostly US-centric (e.g., [6]) or confined to server-client paths (e.g., [19]) or academic sites (e.g., PlanetLab [27]).

There are three metrics of network performance associated with each call: (i) round-trip time (RTT), (ii) loss rate, and (iii) jitter; we do not analyze bandwidth given the low data rate of typical VoIP streams. These network metrics are calculated by the Skype clients in accordance with the RTP specifications [22] and correspond to the average value of each metric over the entire duration of a call. (More detailed network metrics such as transient latency spikes or loss bursts are not reported by the Skype clients.) To understand the characteristics of default Internet routing, this section focuses only on default-routed (BGP-derived) calls while §3 considers relayed (i.e., overlay-routed) calls as well.

2.2 Call quality & Network performance

For a small random fraction of calls in Skype, users label the call quality on a discrete 5-point scale, ranging from 1 (worst) to 5 (best). However, to limit the imposition on users,



(a) RTT (b) Loss (c) Jitter

Figure 2: CDF of observed network performance metrics – RTT, loss rate, jitter. Vertical grey lines show the thresholds for poor network performance.

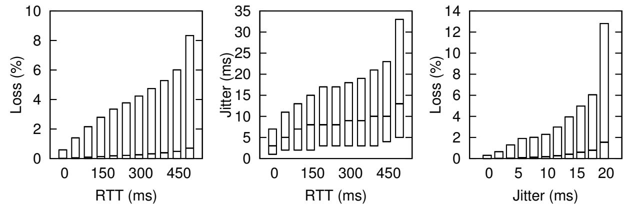
Skype obtains user rating for only a small fraction of the calls. Consistent with the operational practice in Skype, we deem the calls with a rating of 1 or 2 as "poor", and use the fraction of such calls, termed as the *Poor Call Rate (PCR)*, as an empirical metric of user experience. Besides PCR, prior work also has provided analytical models to translate the network metrics into a measure of audio call quality, called the *Mean Opinion Score (MOS)* (e.g., [16]).

In this section, we show that both PCR and MOS are well-correlated with network metrics. Then, we identify suitable *thresholds* for poor call performance on the individual network metrics of RTT, loss and jitter. Since our goal is to understand the impact of network performance metrics, the thresholds keep our focus directly on these metrics.

Does network performance impact user experience? Figure 1 shows the impact of the three network performance metrics (RTT, loss rate, jitter) on the (normalized) user-derived PCR. For each network metric, we bin calls based on their network performance and show the PCR of the calls within each bin. For statistical significance, each bin has at least 1000 samples. The figures show PCR significantly increases with all the three network metrics (correlation coefficients of 0.97, 0.95, 0.91) confirming that user-perceived quality is indeed sensitive to network performance. Interesting, PCR is sensitive to the *entire* spectrum of network metrics. This implies that any improvement in RTT, loss or jitter is likely to improve PCR. MOS (calculated using the model in [16]) also drops with increase in all three metrics (not plotted).

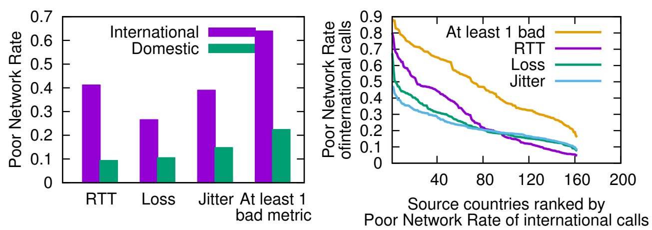
Thresholds of network performance: Figure 2 shows the distribution of network performance experienced by the default-routed calls. A significant fraction of calls (over 15%) occur on paths with RTT over 320ms, or loss over 1.2%, or jitter more than 12ms, which we pick as our *thresholds for poor performance*. These values are in line with literature from industry and standards bodies that recommend one-way end-to-end delay of no more than 150 ms and a packet loss rate of no more than 1% for good call quality [4, 2]. (Note that these thresholds are on the average values over the call's duration during which there may be transient spikes in badness, e.g., loss burst.)

Our focus: Poor Network Rate We define the *poor network rate* (PNR) of a network metric for a set of calls as the fraction of calls whose performance on the metric is worse than the chosen thresholds: RTT $\geq$ 320ms, loss rate $\geq 1.2\%$, jitter $\geq$ 12ms. One of our goals is to reduce PNR of



(a) RTT vs. loss rate (b) RTT vs. jitter (c) Jitter vs. loss rate

Figure 3: Pair-wise correlation between performance metrics. The Y-axis shows the distribution (10^{th}, 50^{th}, 90^{th} percentiles) of one metric as a function the other metric over the same set of calls.



(a) International vs. domestic (b) Countries of one side of a call

Figure 4: International vs. Domestic Calls.

each individual metric (i.e., how often each of them is poor).

However, as there could be correlation between network metrics, improving one metric may increase PNR of another metric. Figure 3 shows the three pair-wise correlations. While the plot is based on an aggregation of data across all calls and paths, the substantial spread suggests at least the possibility that improving one performance metric *could* lead to a worsening of the other metrics. Therefore, we also focus on reducing PNR of three metrics collectively, i.e., how often *at least one* of the metrics is poor.

WAN vs. wireless last hop This work focuses on improving the performance of the WAN path, rather than the last-hop link (e.g., wireless). Previous studies (e.g., [25]) have shown that while the wireless last hop could be a significant contributor to poor call quality even wired clients experience poor calls. Also, as our experiments later in this section show, the PNR for international and inter-AS calls is significantly higher than that for domestic and intra-AS calls. Both these findings suggest that the WAN path does matter, hence our focus here on improving its performance. Calls over poor networks that cannot improve through *any* relaying (see §3) are likely due VIA's limitation of not addressing last-hop issues.

2.3 Spatial patterns in performance

We have seen in §2.2 that user experience is sensitive to poor network performance and that a significant fraction of calls suffers from poor performance owing to default routing. Next, we analyze whether the calls with poor networks share common patterns. This subsection focuses on *spatial* patterns while §2.4 looks at *temporal* patterns.

International vs. Domestic Calls: On all the three network metrics, we see that international calls (between users in different countries) have a higher PNR, i.e., they are more likely to suffer from bad network performance than domestic calls. Figure 4 shows a $2-3\times$ higher PNR on international calls than on domestic calls. The figures also show the fraction of calls with at least one metric being poor (the last pair of bars), where the gap between international and domestic calls is even larger. Though conclusively diagnosing the root cause of bad performance on international calls is hard and beyond the scope of this work[1], the higher PNR for international calls points to the WAN path as the culprit.

Furthermore, Figure 4b zooms into the international calls and classifies them by the country of the callers (source). We see that there is a skewed distribution, with certain countries having a PNR as high as 70% on the individual metrics. The PNR of international calls across the remaining countries drops gradually but half of them still see a non-negligible PNR of $25\% - 50\%$. This suggests that poor network performance is quite widespread, highlighting the suitability of a *globally* deployed overlay network that provides high performance inter-connection between overlay nodes.

Inter-AS vs. Intra-AS Calls: Similar to international calls, calls across ASes are $2-3\times$ more likely to experience poor network performance than those within the same AS domain (figure omitted). This, again, points to the need for enabling alternatives to default routing to improve WAN performance.

Not just a few problematic source-destination pairs: Contrary to our expectation, a few source-destination pairs alone do *not* contribute to a big chunk of the calls to the PNR. Figure 5 shows the fraction of calls that suffer from poor network performance that come from the worst AS pairs, ranked in order of their contribution to the overall PNR. Even the worst 1000 AS pairs together only count for less than 15% of the overall PNR. This means that point solutions that fix a few bad ASes or AS pairs, e.g., informing the AS administrators or the clients directly regarding their ISPs, are not sufficient.

While the above analysis was at the granularity of ASes,

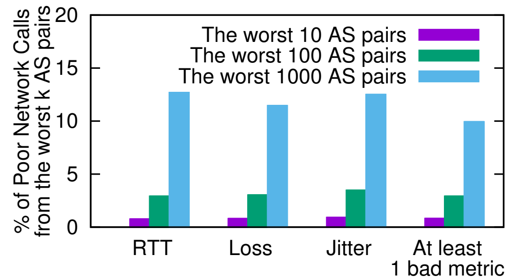
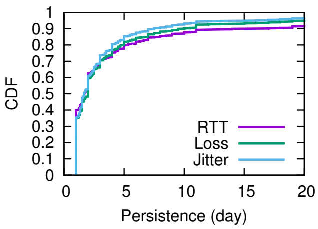
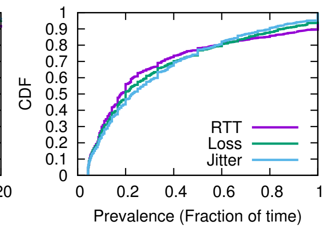


Figure 5: The percentage of calls over poor network conditions that come from the worst k AS pairs; AS-pairs are ranked in descending order of their contribution to total amount of calls with bad performance.

[1] One bias, other than WAN path, that may cause such bad performance is that users tend to use VoIP regardless of its performance, when making international calls, while they have many alternatives, such as cellular, to make domestica calls.



(a) Persistence



(b) Prevalence

Figure 6: Temporal patterns of poor network performance. Figure 6a and 6b show the distribution of the persistence and prevalence of AS pairs having high PNR on each of the network metrics.

we also tested at other, finer granularities (e.g., /24 and /20 prefixes of the caller and callee IP addresses) and found similar results (of not just a few culprits). In fact, for the pairs with sufficient data density at the /24 granularity, we found that performance distributions of the network metrics were similar to those at the granularity of ASes.

2.4 Temporal Patterns in Performance

We now analyze temporal patterns of poor network performance. We perform this analysis by grouping the performance of AS pairs in to 24-hour time windows. We conservatively label an AS pair as having *high PNR* for a specific metric (on a given day) if its PNR on that day is at least 50% higher than the overall PNR of all calls on that day.

Figure 6a and 6b show the distribution of *persistence* and *prevalence* of high PNR AS-pairs. The *persistence* of an AS pair is the median number of consecutive days when it has high PNR. The *prevalence* of an AS pair is the fraction of time it has high PNR. The figures show a highly skewed distribution with 10% − 20% AS pairs always having high PNR, while 60% − 70% AS pairs have poor performance for less than 30% of time and lasting no longer than one day at a stretch. Such a skewed distribution suggests that instead of statically configuring the system to improve performance for only the (relatively few) most prevalent and persistent AS pairs, we need to dynamically decide if a call should use default Internet routing or be relayed.

2.5 Key observations and implications

The key observations from this section are:

1. *Network performance matters.* User experience of calls is impacted by even small changes in network metrics.
2. *Wide-area* communication, such as international and inter-domain calls, are more prone to bad network performance, and have a large room of improvement.
3. Calls suffering from poor networks are spread *spatially* and *temporally*. Most calls with poor network performance are not from a handful of source-destination AS pairs. And most source-destination pairs only experience high PNR for a relatively short period of time.

These observations motivate the need for a network overlay (Observation 1) that provides better paths with a *global* footprint of overlay nodes (Observation 2), and the need to choose routes *selectively* and *dynamically* (Observation 3).

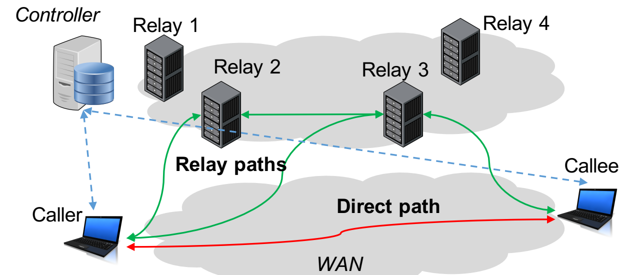


Figure 7: VIA architecture with relay nodes at globally distributed data centers. A call can either take “default path” (red) or a “relay path” (green).

3 Approach and Potential of VIA

In this section, we present the approach of VIA, a managed overlay architecture that consists of relays hosted at globally distributed data centers and a centralized controller making dynamic relay selection (§3.1). Then, §3.2 quantifies the potential of VIA to improve calls with poor network performance that were characterized in §2. As a preview of our results, we find that an oracular scheme, with careful decisions made with regard to relaying, could help improve the network metrics for calls by 30%-60% at the median and the PNR (poor network rate) on these metrics by over 30%.

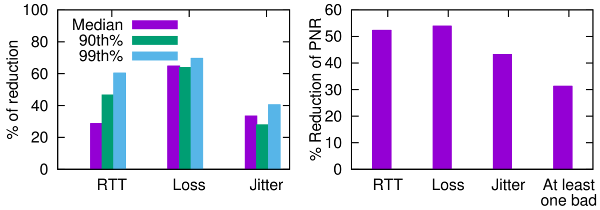
3.1 VIA Architecture

Figure 7 presents the VIA architecture that consists of relay nodes placed at globally distributed datacenters, such as those run by Amazon, Google, and Microsoft. Indeed, VIA’s architecture bears similarities to those used by Google Hangouts and Skype [36], but with a key difference — today, the relays are typically used to provide connectivity between any two clients, while VIA is engineered to *explicitly* optimize network performance and call quality.

Each call can take either the “*default* path” (red arrow) or a “*relayed* path” (green arrows) that routes the traffic through one or more relay nodes in the DCs. Relayed paths could include a single relay to "*bounce* off" traffic or a pair of relays to enable traffic to "*transit* through" the private backbone of the managed overlay network.

In our study, we use all the relay nodes operated by Skype. They are all located in a single AS (so all inter-relay paths are within a private WAN) but spread across many tens of datacenters and edge clusters worldwide. We assume the caller (or callee) can reach these relays by explicitly addressing the particular relay(s). The network path between a relay and a client is determined by BGP.

When establishing a call, after the caller signals its callee, both the caller and callee contact a *controller* (Figure 7) to determine whether they should use the direct path or a relayed path, and, in case of the latter, which relays they should use. The controller makes this decision based on the performance measurements from historical calls and policy con-



(a) Performance distribution (b) Poor Network Rate

Figure 8: Potential improvement of VIA.

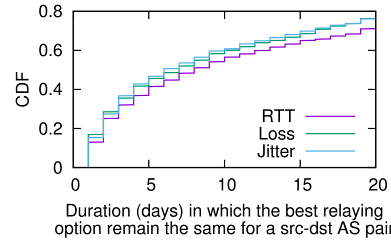


Figure 9: Distribution of how long the best relaying option (picked by oracle) lasts. The optimal relaying options for 30% of AS pairs last for less than 2 days.

straints (such as those based on relay budget or current load), to be described in §4. To aid in this process, Skype clients routinely push the network metrics derived from their calls, to the controller. As §2 motivated, the controller dynamically updates its decisions using the latest measurements.

The controller does not need to directly monitor the relay nodes because their performance (including degradation and failure) would be reflected in the end-to-end measurements made by clients who use the relays. To avoid overloading the controller, each client could cache the relaying decisions and refresh periodically though we do not consider this here (see more discussion of implementation issues in §7).

3.2 Potential Relaying Improvement

We report results from the *first large-scale study of the potential benefits of relaying on Internet telephony*, using the dataset of VoIP calls from the Skype service.

We quantify the potential gains of VIA, using an "oracle" control logic, which enjoys the benefit of foresight. For each call between a source-destination pair, it has knowledge of the average performance of each *relaying option* on a given day. As shown in Figure 7, a relaying relaying option could be either the default (direct) path, a bouncing relay path, or a transit relay path. For each source-destination pair, the oracle picks the relaying option that has the best average performance (i.e., lowest RTT, loss rate, or jitter) for this source-destination pair on this day[2]— either a relay path or the direct path. We also have information from Skype on the RTT, loss and jitter between their relay nodes, which we use in estimating the performance of a transit relay path.

The oracle makes two simplifying assumptions: (1) there are no load restrictions on the relays or the network backbone, and (2) the performance measurements of each relaying option are unbiased samples of its actual performance. In §4.7, we will relax the first assumption by introducing a budget constraint on fraction of calls being relayed.

Gains from oracle approach: Figure 8 shows the improvement (i.e., reduction) in the values of RTT, loss and jitter individually as well as the PNR (defined in §2.2). Specifically, if a statistic goes from b to a, we define the *relative* improvement as $100 \times (\frac{b-a}{b})$, which lies between 0 and 100.

[2]Picking a day's granularity gives us sufficient samples for most of the relaying options. Nevertheless, for the small fraction of source-destination pairs for which we had sufficient samples on a timescale of minutes, we found that the oracle still had a significant benefit.

The oracle can help reduce RTT, loss and jitter by 30%-60% at the median (Figure 8a). Reduction at the tail, which is of particular significance in the context of user-perceived quality, is nearly 40%-65% with the oracle's choice of relaying. All this translates to a healthy reduction in the PNR on each of RTT, loss, and jitter (Figure 8b), which, in turn, would help considerably in terms of user satisfaction; for instance, PNR on loss rate is reduced by 53%.

We also analyze the reduction in PNR when the three metrics are considered *together*, i.e., improving from a situation where at least one of the metrics is poor to a situation where *none* of the three is poor (i.e., RTT is under 320ms, loss is under 1.2%, and jitter is under 12ms), while still optimizing for RTT, loss and jitter individually. This helps us understand how correlated the improvement in the three metrics is. It is encouraging that even with such a strict stipulation, we can achieve substantial reduction in PNR; for instance, when selecting the relays that reduce average loss rate, we can reduce the fraction of calls with bad performance on at least one metric by over 30% (Figure 8b, right-most bar).

Need for dynamic relay selection: Whether the control logic should select relay dynamically or statically depends on how often the relaying decisions need to be updated. Figure 9 shows the distribution of the median duration during which the oracle picks the same relaying option for a source-destination AS pair. We see that the optimal relaying option for 30% of AS pairs lasts for less than 2 days, and only 20% of AS pairs have the same optimal relay option for more than 20 days. This, together with the observation on the relatively low persistence of poor performance (Figure 6), suggests that the relay selection should be done *dynamically*, rather than statically.

4 VIA Relay Selection

Having shown that, with the benefit of an oracle, relaying through VIA could provide significant gains, we now turn to devising a *practical* algorithm for relay selection. We begin by formulating the problem of relay selection. We describe two classes of strawman approaches — *predictive* and *exploration-based* — and highlight key limitations of both classes. We then present the core intuition behind our relay selection algorithm, called *prediction-guided exploration* and then describe the detailed design.

4.1 Problem formulation

At a high-level, our goal is to assign each call to a particular relaying option as discussed in §3.1. Recall that a relaying option can be using the default path, using a specific one-hop relay node (i.e., bouncing relaying), or using a specific pair of relay nodes (i.e., transit relaying). Let C denote the set of calls we want to optimize and let R denote the set of available relaying options. We use $c \in C$ and $r \in R$ to denote a specific call and relaying option, respectively. Let $Q(c,r)$ denote the expected value of a network metric for c when using r (a smaller value is better). We assume that the relaying decisions for calls are independent in that the performance of a call is not impacted by the relaying decisions made for other calls.

The goal of VIA is to *assign* optimal relaying options for each $c \in C$. Let $\mathsf{Assign} : C \rightarrow R$ denote the assignment function output by some algorithm and let $\mathsf{Assign}(c)$ be the relaying option assigned for call $c \in C$. Formally, our objective is to find the optimal assignment

$$\arg \min_{\mathsf{Assign} \in R^C} \sum_{c \in C} Q(c, \mathsf{Assign}(c))$$

This is a minimization problem because a lower value is better for each of our network quality metrics Q.

4.2 Strawman approaches and limitations

At a high level, we can consider two classes of approaches for choosing the optimal assignment of relaying options to upcoming calls:

1. *Exploration-based:* One approach is to set aside a fraction of the calls for measurement-based exploration of the performance of each possible relaying option for every source-destination pair. For instance, for every AS-pair and every possible relaying option r, we will explicitly use some of the calls to explore the option and measure the performance, $Q(c,r)$.
2. *Prediction-based:* An alternative to the exploration-based approach is to use the recent history of observed call performance. Suppose, VIA has available as input call records with measured performance H. Then, we can use suitable prediction algorithms to predict the performance $Q(c,r)$ for every combination, and select the option that has the best predicted performance.

Unfortunately, we observe in practice that both these classes of approaches have very poor accuracy in predicting $Q(c,r)$. This ultimately results in a poor assignment strategy and poor call quality. There are two key reasons:

- First, there is a fundamental problem because of *skew in data density.* Specifically, there is a substantial difference in the number of call samples available across different source-destination pairs, both for the direct path and for the various relayed paths. This variability arises because of the large space of choices: N end-points and M relay strategies lead to $O(N^2M)$ choices. Furthermore, certain end-points inherently make/receive fewer calls, yielding fewer samples.
- Second, there is *inherent variability* in the observed performance. Consequently, to estimate $Q(c,r)$, we need a significant number of samples before the empirically observed values can converge to the true values.

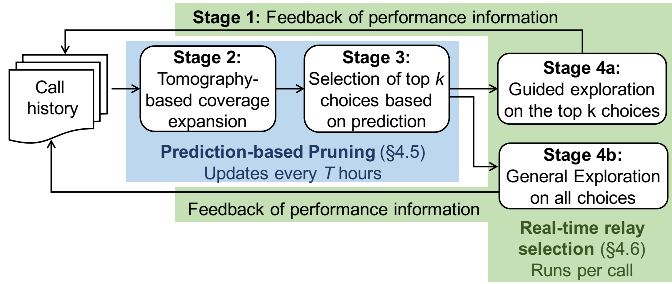


Figure 10: Overview of VIA relay selection based on prediction-guided exploration.

The skew and the variability make prediction inaccurate and exploration ineffective and/or expensive (in terms of the effort to be expended).

4.3 Intuition behind our approach

The key intuition behind our approach is the empirical observation that even though the prediction-based approach may not predict the optimal choice, the optimal is likely contained in the top few among its predictions. In other words, if we look at the top-k choices (those who have the best predicted performance) suggested by the prediction-based approach, the optimal choice will likely be a member of that set, even though it is a small subset of all possible relaying options R.

We can exploit this observation to *prune* the search space for our exploration step. That is, the exploration approach does not need to blindly explore the set of all possible strategies R, but instead can focus on a much smaller set of top-k predictions. We refer to this as a *prediction-guided exploration* approach. The top-k pruning is not to be confused with a similar machine learning problem which seeks to find k best arms (e.g., [10]). In contrast, we care more about the best relaying option – our top-k candidates may have bad options, but the best relaying option is very likely to be among them, and can be found by exploration techniques.

4.4 Overview of the VIA relay selection

Figure 10 depicts the main stages in VIA, and Algorithm 1 shows the pseudocode. In a nutshell, the logical stages are:

1. gathering performance information from call history,
2. using network tomography to expand the coverage of the information from call history,
3. using the (expanded) history information to predict performance and prune all but the most promising top-k relaying options, and

```
Input: A set of calls C to be assigned to the set of relaying
       options R, Set of historical calls H
Output: A relay assignment Assign where Assign(c) ∈ R is
        the relay option assigned for call c ∈ C
1  /* Stage 2: Tomography-based performance
      predictor trained from H                               */
2  Pred ← BuildPredictor(H);
3  Assign ← ∅;
4  for (s, d) do
5      /* Stage 3: Pick Top-k candidates based on
          history-based prediction.                          */
6      TopK ← GetTopK(s, d, R, Pred);
7      for c ∈ C do
8          if RandomFloat(0, 1) < ε then
9              /* Stage 4a: Explore the Top-k
                  candidates                                 */
10             r ← Explore(c, s, d, TopK, Assign, Pred);
11         end
12         else
13             /* Stage 4b: Randomly explore all
                  relaying options                           */
14             Assign(c) ← Random(R);
15         end
16         Assign(c) ← r;
17     end
18 end
19 return Assign;
```

Algorithm 1: Relay selection algorithm of VIA

4. perform exploration-exploitation on the top-k relaying options as well as all relaying options using multi-armed bandit (MAB) techniques.

Finally, the observed performance of each call will be stored in call history, i.e., fed back to stage 1. Stages 2 and 3 (shown in light blue) are performed at a periodicity of T hours (by default 24 hours), i.e., the pruned list of candidate relaying options are refreshed every T hours. And stages 1 and 4 (shown in light green) are performed on a per-call basis. We discuss these stages in the sub-sections that follow.

4.5 Prediction-based pruning

Using call history data, VIA proceeds to predict, with confidence intervals, the performance between a source-destination pair over each relaying option: the direct paths and each of the relays, transit relays as well as bouncing relays.

Expanding coverage by network tomography: The call history tells us about the performance of network paths that were actually used. As there is skew in call distribution, there might be "holes", i.e., no call history for the network path between a source-destination pair through a specific relaying option. Can we learn about the performance of these network paths?

If we knew the performance of the individual network segments (e.g., client to relay) that comprise an end-to-end path, we could compose these to estimate the performance of the path. In principle, measurements of the individual network segments could be made by the relays themselves. However, the relays in our system were only designed to forward Skype traffic and we were not in a position to add new functionality to these relay nodes (and potentially impose an additional overhead). The situation is not unlike that in ISP networks where direct measurement of network links, while easy in principle, is challenging in practice, leading to indirect approaches for estimating such metrics as the link performance and the traffic matrix [37].

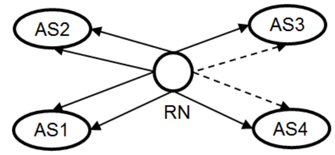


Figure 11: Path stitching in VIA to estimate performance through relay RN. Solid lines represent historical call samples that we use to predict performance between AS3 and AS4 (dotted line). $\mathbf{RTT}_{\mathbf{AS3}\leftrightarrow\mathbf{AS4}} = \mathbf{RTT}_{\mathbf{AS1}\leftrightarrow\mathbf{AS4}} + \mathbf{RTT}_{\mathbf{AS2}\leftrightarrow\mathbf{AS3}} - \mathbf{RTT}_{\mathbf{AS1}\leftrightarrow\mathbf{AS2}}$.

Network tomography provides an alternative. By combining end-to-end measurements across several, partially-overlapping paths, network tomography can help estimate the performance of each network segment. Then, by stitching together the estimates for the individual segments, we can estimate the performance of a path not seen before.

Figure 11 shows a simple example of how network tomography expands coverage. We use linear tomography, and apply it to individual metrics that compose linearly (e.g., RTT) or can be linearized (e.g., jitter and packet loss rate, under the assumption of independence across network segments [11]).

We model BuildPredictor as a network tomography problem. Given a relay path that uses relaying option r and between source AS s and destination AS d, our tomography algorithm models it as a path consisting of two segments: a segment between s and r and a segment between d and r. Modeling network end-points on AS level is a pragmatic trade-off: it gives us sufficient data on many source-destination pairs, and still produce significant improvement (see §5.5 for comparison between different granularities). The prediction algorithm can work at a finer granularity (e.g., /24 IP prefix) when more data are available.

The resulting Pred module (Algorithm 1, line 2) predicts for a source-destination pair (s and d) both the mean performance $\mathsf{Pred}_{mean}(s,d,r)$ for a specific relaying option r, and its standard error of mean (SEM) $\mathsf{Pred}_{sem}(s,d,r)$. [3] Based on these, Pred estimates both the lower and higher 95% confidence bound by $\mathsf{Pred}_{lower}(s,d,r) = \mathsf{Pred}_{mean}(s,d,r) - 1.96\mathsf{Pred}_{sem}(s,d,r)$ and $\mathsf{Pred}_{upper}(s,d,r) = \mathsf{Pred}_{mean}(s,d,r) + 1.96\mathsf{Pred}_{sem}(s,d,r)$.

Pruning to get top-k choices: Pruning does not necessarily narrow down to the single best relaying option. However, we see that the best relaying option is often among the top-k predicted options for a small value of k. For instance, the probability of the option with the minimum RTT being included even in top three or four ($k = 3$ or 4) is $60\% - 80\%$

[3]The standard error of mean is the standard deviation of a sample mean's estimate of the population mean. It tends to zero as the number of samples grows.

Input: Source AS s, destination AS d, all relaying options R, and predictor Pred
Output: Top k relaying options $TopK$ for calls between s and d

Function GetTopK(s, d, R, Pred)
  `/* Initializing variables */`
  $TopK \leftarrow \emptyset$; $Remained \leftarrow R$; $h \leftarrow \infty$;
  while *true* **do**
    $r \leftarrow \arg\min_{r \in Remained} \mathsf{Pred}_{upper}(s, d, r)$;
    if $\mathsf{Pred}_{lower}(s, d, r) > h$ **then**
      Break;
    end
    else
      $h \leftarrow \mathsf{Pred}_{high}(s, d, r)$;
      $Remained \leftarrow Remained \setminus \{r\}$;
      $TopK \leftarrow TopK \cup \{r\}$;
    end
  end
  return $TopK$

Algorithm 2: Getting the top-k choices based on prediction.

as against just 29% if we were to pick only the option with the predicted minimal RTT ($k = 1$). Therefore, we adopt the approach of using our predictor to pick the top-k relaying options and use that for *guided exploration*.

Instead of using a fixed value of k, VIA *dynamically decides* k based on the lower and higher confidence bounds for each relay r on the particular source-destination pair s and d. Algorithm 2 shows the pseudocode. Specifically, we define top-k to be the minimal set of relaying options such that the lower 95% confidence bound ($\mathsf{Pred}_{lower}(s, d, r)$) of any relay option not in the top-k is higher than the upper 95% confidence bound ($\mathsf{Pred}_{upper}(s, d, r)$) of any relay option in the top-k. (Recall that the lower the value of a network metric, the better it is.) In other words, we are very sure that any relay option that is not included in the top-k is worse than any that is. For instance, the probability of the option with minimal RTT being included in such top-k is over 90%.

4.6 Relaying selection in real time

Exploring the top-k choices for each source-destination pair (Explore of line 10 in Algorithm 1) is the classic multi-armed bandit problem, where each of the relaying options is an "arm" of the bandit and the network performance obtained is the "reward". While bandit selection is a much studied problem, doing so under high-variance and dynamically changing performance distributions (i.e., rewards) of the bandits, and also limited budget for each bandit, requires interesting adaptations, as outlined below.

In its basic form, relay options are selected by an *exploration-exploitation* process, where a fraction of calls is assigned to explore different relay options (ϵ-greedy) and the rest exploit the best decision[4]. As briefly mentioned earlier, standard exploratory approaches are slow to converge (Figure 10). As

[4] A similar exploration-exploitation process could be invoked on per-packet basis within the same call. However, this would require packet-level control, which is out of the scope of this paper.

Input: A call c, source AS s, destination AS d, top k relaying options $TopK$, relay assignment of history calls Assign, and predictor Pred
Output: Relay selection of c

Function Explore$(c, s, d, TopK, \mathsf{Assign}, \mathsf{Pred})$
  `/* Initializing variables */`
  $ucb_{min} \leftarrow \infty$; $r_{top} \leftarrow$ null;
  `/* To avoid outliers, we do not use maximum performance value as normalizor w. */`
  $w \leftarrow \frac{1}{|TopK|} \sum_{r \in TopK} (\mathsf{Pred}_{upper}(s, d, r))$;
  `/* Following is the standard UCB1, except for the normalization scheme. */`
  $T \leftarrow |\mathsf{Assign}| + 1$;
  for $r \in TopK$ **do**
    $C_r \leftarrow \{c' | \mathsf{Assign}(c') = r\}$;
    `/* Q is the quality function. */`
    $ucb \leftarrow \frac{1}{w|C_r|} \sum_{c' \in C_r} Q(c', r) - \sqrt{\frac{0.1 \log T}{|C_r|}}$;
    if $ucb < ucb_{min}$ **then**
      $r_{top} \leftarrow r$; $ucb_{min} \leftarrow ucb$;
    end
  end
  return r_{top}

Algorithm 3: Exploring the top-k candidates in real time using modified UCB1.

we will show in §5, they often fail to select the best decision. This is because exploring in presence of high variability requires a lot of samples, which is infeasible due to data sparseness and skew.

We build off the UCB1 algorithm [8]. Algorithm 3 shows the pseudocode. UCB1 is well-suited for our purpose because it does not require explicitly specifying the fraction of samples for exploration. Instead, it transparently combines both exploration as well as its exploitation decisions.We make two modifications to make it work well in our context.

1. UCB1 normalizes rewards (i.e., performance) from each bandit (i.e., relay option) to be between 0 and 1. In our situation, however, normalizing based on the full range of values of each performance metric is problematic due to the large variance in distribution of the metrics (e.g., unusually large RTT). Normalizing all values based on such a wide range leads to poor decisions, since the difference between values in the common case become hard to discern. Instead, we normalize the rewards by dividing them by the average of upper 95% confidence bounds ($\mathsf{Pred}_{upper}(s, d, r)$) of the top-$k$ candidates.

2. The top-k pruning in §4.5 is a function of only the samples explored. Therefore, to avoid being blindsided by dynamically changing performance distributions, VIA also sets aside ϵ fraction of calls to random relays (outside of the top-k) for *general* exploration. This step is not required in traditional exploration-exploitation techniques as they assume the reward (performance) distribution of each bandit (relay option) is static, which may not hold in our context.

4.7 Budgeted relaying

We extend VIA's relaying decision to consider budget constraints: specifically that the fraction of calls being relayed must be less than a certain limit, B (e.g., 30%). While such an overall budget on relayed calls is simple, in general it may also be of interest to consider other budget models, such as per-relay limits or bandwidth cap on call-related traffic.

VIA utilizes the budget using a simple extension to the heuristic in §4.6. It decides to relay a call only if the benefit of relaying is *sufficiently* high. If the overall budget for relaying calls is B percent, a call should be relayed only if the benefit of relaying it is within the top B percentile of calls. VIA uses historical call information (of relaying benefits) to keep track of the percentiles. It decides to relay a call only if the expected benefit is above the B^{th} percentile benefit.

5 Evaluation

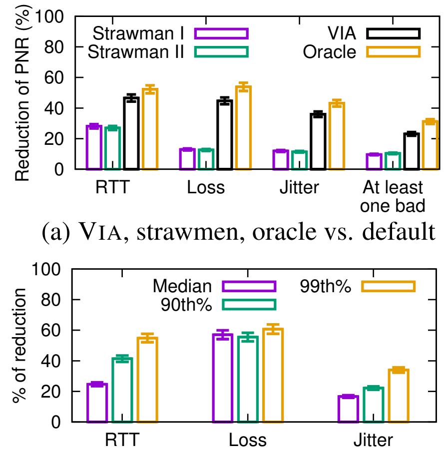
In this section, we show that VIA can significantly improve performance on network metrics. Specifically, we show that:

- VIA achieves substantial improvement on all network metrics — 20-58% reduction on median, 20-57% on 90th percentile and 35-60% on 99th percentile. And VIA reduces PNR by 39-45% for the individual metrics, and by 23% when PNR is computed on an "at least one bad" metric basis.
- VIA achieves close-to-optimal performance under budget constraint, even when VIA relays any calls that may have benefits whenever there is budget. Moreover, when VIA only relay calls that have higher potential benefits (i.e., the modification in §4.7), we see even higher improvement within the same budget.
- The improvement rises when the relay decisions are made at a finer spatial granularity and more dynamically. However, we start to see diminishing gain after the decisions are made per AS pair and daily.

5.1 Methodology of data-driven evaluation

We perform data-driven simulations based on 430 million Skype calls (§2.1). The calls are replayed in the same chronological order as in the trace thereby allowing VIA to gain knowledge as it goes along. We assume that when a call is assigned to certain relay option, its performance would be the same as that of a call which is randomly sampled from the set of calls between the same AS pair through the same relay option. Therefore, we focus on AS pairs and dates where there are at least 10 calls on at least 5 relay options [5] (after which, 32 million calls remain in the evaluation). We acknowledge this methodology makes the same simplifying assumptions as the oracle (§3.2). To quantify the confidence in the results, we also add error bars (of standard error of mean) to the graphs. Note that even with the aggregation, we used *distribution* (e.g., mean, percentiles) of the metrics and not per-call values for evaluation.

[5] Otherwise, selecting relays from a handful of candidates would be trivial. And in a complete deployment, the potential relay to choose from are much more than five.



(a) VIA, strawmen, oracle vs. default

(b) VIA improvement on percentiles

Figure 12: Evaluation of overall improvement of VIA.

This section shows how much VIA can reduce PNRs (fraction of calls having poor performance on the individual network metrics or on the "at least one bad" metric), compared with the oracle approach and a strawman, such as using the default paths for all calls ("default strategy").

5.2 Overall improvement of VIA

PNR reduction: Figure 12a shows the PNR reduction of VIA over default strategy (always using default paths), and compares it with the PNR reduction of pure prediction-based selection, based just on history (Strawman I), pure exploration-based selection without any pruning of the options up front (Strawman II), and oracle. Across all three performance metrics, we see that VIA achieves close-to-oracle performance and significantly outperforms both the default strategy and the two strawman approaches. The strawman approaches yield much less improvement, which confirms the inefficiency of the pure predictive strategies and the pure exploratory strategies (§4.2).

Improvement on percentiles: Next, Figure 12b shows the VIA improvement over default strategy on the performance percentiles of three network metrics. We first calculate the percentiles of performance of each strategy and calculate the improvement between these percentiles (which avoids the bias of calculating improvement on each call). We see that VIA has improved performance on both median (by 20-58%) and the extreme tail (by 20-57% on 90th percentile), which shows VIA is able to improve the performance of a wide spectrum of calls.

5.3 Component-wise analysis

Prediction accuracy of relay-based tomography: As a first step, VIA uses relay-based tomography (§4.5) to predict the performance each relaying option. We evaluated the accuracy of tomography-based predictions on the different metrics and found that on 71% of calls, the predicted performance is within 20% from the actual performance. However, for 14% of the calls, the error can be $\geq$ 50%. This non-negligible prediction error explains the poor performance of

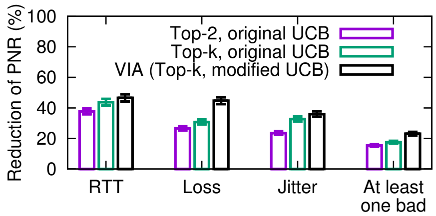
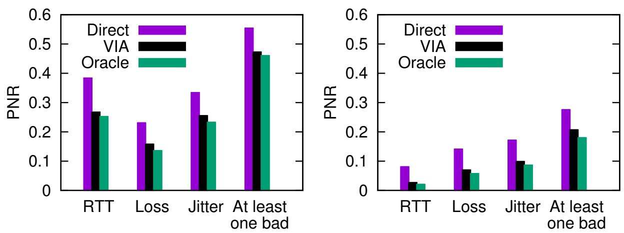
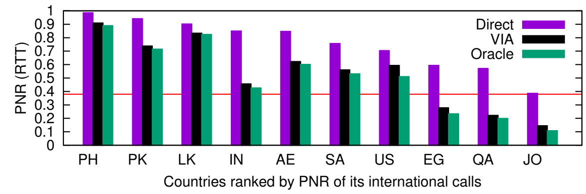


Figure 13: Comparing various guided-exploration strategies.

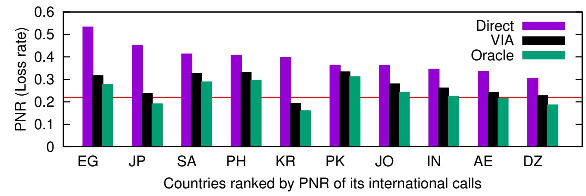


(a) International (b) Domestic

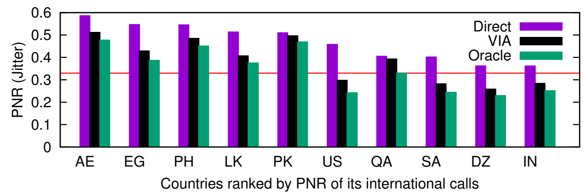
Figure 14: VIA improvement on international and domestic calls. We also have similar observation regarding inter-domain and intra-domain calls.



(a) PNR of RTT



(b) PNR of loss rate



(c) PNR of jitter

Figure 15: Dissecting VIA improvement on PNR by country of one side. There is a substantial diversity on VIA improvement across different countries.

Strawman I (pure prediction-based) that we have seen in Figure 12a, and also motivates real-time exploration.

Benefits of prediction-guided exploration: As discussed in §4, VIA is not a simple combination of prediction and exploration approach. First, instead of picking a fixed number top candidates, VIA pick top candidates by taking variance of prediction into account. Second, instead of using the original UCB1 algorithm, which assumes a normal distribution of rewards, we adopt a different way to normalize values to cope with performance outliers. Figure 13 quantifies the incremental contribution of both modifications on PNR of the three metrics. It shows that each modification makes a significant contribution to VIA's improvement. With the "at least one bad" metric, picking top k and using the normalized reward, instead of just the top 2, raises the improvement in PNR from 15% to 24% (or loss rate from 26% to 44%).

International vs. domestic Figure 14 compares PNR of international and domestic calls under strategies of default, VIA and oracle.We see significant improvement of VIA on both international and domestic calls, while international calls have a slightly higher magnitude of improvement than domestic calls. This can be explained by the fact that relaying has limited benefits when the bottleneck is the last-mile ISP or the last-hop connection.

Benefits by countries: Figure 15 further dissects the improvement of VIA by specific countries of one side. It shows that many countries have a much higher PNR than the global PNR (shown by the horizontal red line), and that the performance of VIAis closer to the oracle than to the default for most of these countries.

5.4 Budget constraints

Being able to use relays judiciously within a budget for relayed calls is an inherent requirement in the context of managed overlay networks such as VIA. Here, we define budget as the maximum fraction of calls being relayed. We only impose an overall budget, not a per-relay one. Figure 16 shows the impact of budget on PNR (of at least one bad metric) of three strategies: oracle, budget-unaware VIA and budget-aware VIA. The budget-unaware VIA, which selects relays based on Algorithm 1, will relay calls whenever there is potential benefit of doing so, without taking into consideration the overall budget of relaying. Therefore, there is a risk of the budget getting used up by calls with only small benefit. In contrast, budget-aware VIA (§4.7) relays a call only when the benefit is larger than a threshold, which depends on the actual budget. That means calls with minimal benefit will not be relayed, saving resources for the calls that would benefit the most by relaying. From Figure 16, we see that the budget-aware VIA (§4.7) can use budget much more efficiently than the budget-unaware VIA. Also, budget-aware VIA can achieve about half of the maximum benefit (i.e., when budget is 100% of calls) with a budget of 0.3 (i.e., only relying 30% of calls).

5.5 Sensitivity to relaying parameters

Next, we show how the benefit of VIA can be affected by various aspect of the managed overlay, including how frequently the relay decisions are made and which type of relays (transit, bouncing or both) are used.

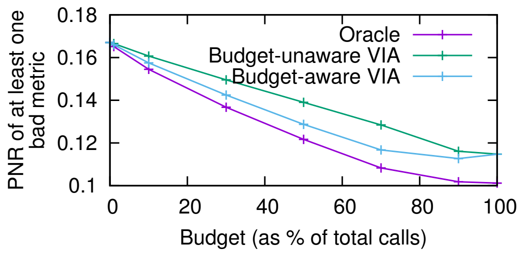


Figure 16: Impact of budget constraint on improvement of VIA.

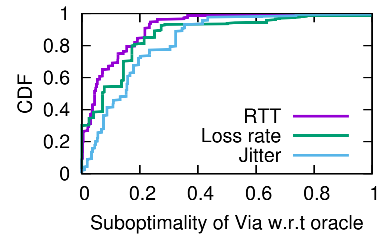


Figure 18: Results of the real-world experiment. Each curve shows the CDF of sub-optimality of VIA's performance on each call.

Decision granularities: We show performance improvement as a function of the spatial and temporal granularity at which VIA operates. First, to show the impact of spatial granularity, Figure 17a fixes the temporal granularity to running stage (2) and (3) of VIA every 24 hours, i.e., $T = 24$ hours (§4.4) and compares the PNR if different relay options could be selected for calls in different spatial granularities. For fair comparison, the PNR are calculated based on the same set of calls.

We see two consistent trends. First, making decision at granularities coarser than a per AS pair results in a smaller reduction in PNR. For instance, different ISPs within a country have different peering relationships, and thus may have different optimal relay options, but such opportunities will not be exploited when making decision per country. (2) Making decisions on finer granularities does not help much, though for a different reason. At finer granularities, the coverage becomes much smaller, which make VIA unable to predict many potential relay options. In future work we hope to analyze a much larger data set In Figure 17b, we see a similar pattern when comparing PNR of different temporal granularities, i.e., different values of T (§4.4).

Relay deployment: Figure 17c shows reduction of PNR when a subset of (least used) relays is excluded. We see that the contribution of benefits from different relay nodes are highly skewed. The effect of having the 30 least used ones given that we already have the 40 most used ones is much less than having the 30 most used ones. This also suggests that new relays should be deployed carefully in order to avoid wasting resources.

Transit vs. bouncing relay: Finally, we find that also using transit relaying (i.e., using inter-DC connection between the ingress and egress relays as part of the path) usually results in higher improvement on PNR than only using bouncing relays (i.e., using one relay node to bounce off traffic). On AS pairs which have used both bouncing and transit relays, we see 50% lower PNR when both transit and bouncing relays are available than when transit relays are excluded. We also find that VIA sends about 54% calls to bouncing relays, 38% to transit relays, 8% to default paths, with a marginal difference in the distribution across network metrics.

5.6 Real-world controlled deployment

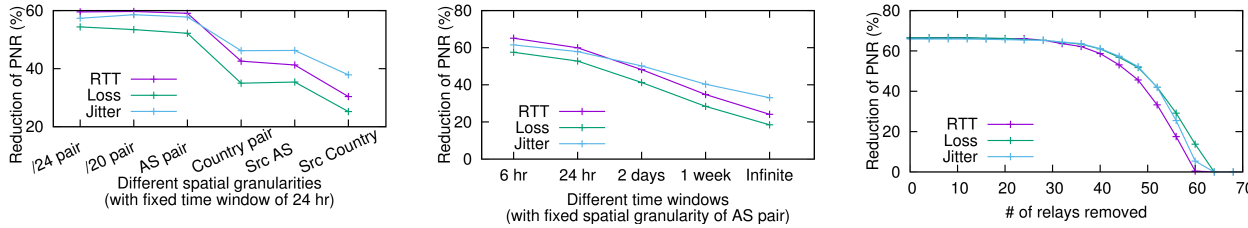
We implemented and deployed a prototype containing the relevant components of VIA at a small scale using modified Skype clients and using Skype's production relays. The central controller of our prototype (Figure 7), deployed on the public Microsoft Azure cloud, aggregated performance measurements from instrumented Skype clients and implemented the relay selection algorithm. The instrumented Skype clients contacted the controller to decide which of the relays of Skype, if any, to use for their calls. We deploy the instrumented client on 14 machines across Singapore, India, USA, UK and Sri Lanka. Overall, we required minimal modifications to the Skype client.

The controller also orchestrated each client to make calls to the other instrumented Skype clients. In total, it creates around 1000 calls between 18 caller-callee pairs. Specifically, it instructed each caller-callee pair to make (short) back-to-back calls using 9-20 different relaying options, 4-5 times each. Since our testbed is at a small scale, such back-to-back calling provides us with high density performance samples between source-destination pairs through many different relays. We use these samples to perform a controlled experiment on VIA's relaying heuristic with accurate ground truth. The results are shown in Figure 18, where each curve shows the CDF of sub-optimality of VIA's performance on each call, defined by $\frac{Perf_{\text{VIA}} - Perf_{oracle}}{Perf_{oracle}}$. We found that VIA's relaying decision is within 20% worse than an oracle's performance for more than 70% of the calls.

6 Related Work

Overlay routing: Overlay networking has been explored in a variety of contexts, such as virtual private networks (VPNs) and multicast [20, 30, 9]. Of interest to us here is work focused on overlay routing with a view to improving performance [32, 7]. This work showed that performance in terms of network metrics such as delay and packet loss, and also reliability, could be improved by using an overlay path that traverses well-chosen way-points.

Despite this promise, such overlay routing for performance gains has not seen much adoption in practice, for several reasons including the last-mile performance bottlenecks encountered in using client nodes as peers and the policy issues involved in turning stub networks (e.g., university campus networks) into de facto transit networks. Perhaps most importantly, these efforts involved building up overlay networks from scratch, both in terms of physical infrastructure and network probing, which limited their scale.



(a) Impact of spatial granularity (b) Impact of temporal granularity (c) Impact of relay deployment

Figure 17: Sensitivity analysis of VIA improvement. Figure 17a and 17b compares PNR under different control granularities. Figure 17c shows PNR when some of the (least used) relays are excluded.

Our work revisits the idea of overlay routing in the context of (a) global-scale managed networks, so the global infrastructure already exists and need not be built up from scratch, and (b) a large-scale interactive real-time service, Skype, which provides both a compelling need for improving performance and (passive) measurements to obviate the need for active network probing.

Evolution of AV Conferencing Services: The architecture of audio-video conferencing services has been evolving, with a trend towards leveraging cloud resources. A case in point is Skype, which started off with a peer-to-peer approach to NAT and firewall traversal, with some well-connected clients with public IP addresses serving as super-nodes [26]. However, in the recent years, Skype has moved to a hybrid model [36], with some super-nodes hosted in the cloud [3]. It has been reported that Google Hangouts uses relays in the cloud for all calls, and moreover also has streams traverse the cloud backbone from one relay to another [36].

Our work is in line with these trends, but focused on performance rather than NAT/firewall traversal. Also, since we focus on managed networks, being selective in which streams are routed via the cloud is crucial in our context.

CDN Server selection: Optimal server selection is a much-studied problem, especially in the context of a large-scale replicated infrastructure such as content distribution networks [34, 33]. The main considerations in the selection process are typically proximity of the client to replicas and the load on the replicas. The main distinction of our work is our focus on client-to-client communication, which means that relay selection needs to focus on end-to-end performance rather than just between the cloud edges and the client.

Internet Performance Prediction: There is a large body of work on Internet performance prediction [21, 28, 27], with a focus on metrics such as bandwidth, delay, and packet loss rate. The general approach is to probe the network selectively, at chosen times and along chosen paths, and then to use the measurements to either embed the network nodes in a coordinate space [17] or estimate the performance of network segments using network tomography techniques [11]. Since we have access to network metrics for a large volume of calls, our work focuses on leveraging this data rather than performing active measurements.

Measurement studies: Over the years, there have been a number of measurement studies of large Internet services, including web sites [29], CDNs [31], and video-on-demand streaming [15, 18]. There have also been studies of audio-video conferencing by working outside the system, say by running active measurements to Skype super-nodes [35] or sniffing traffic in modest-size deployments [36]. To our knowledge, ours is the first study of a commercial VoIP service at scale by directly working with end-to-end performance metrics recorded by the communicating peers themselves.

Estimating VoIP Quality: Several models have been proposed and studied for estimating VoIP quality, typically the Mean Opinion Score (MOS), based on network performance metrics [16, 14, 35, 12, 13]. These models vary in the particular network metrics and codecs they consider. In Section 2.2, we used the model proposed in [16], which is based on the E-Model defined by the ITU [1].

7 Discussion

Active Measurements

Instrumenting relay servers

Cost of centralized control in VIA VIA uses a (logically) centralized architecture. On one hand, it leads to additional measurement updates and control messages sent between clients and controller, but the overhead might be acceptable given that the update/control messages would be one per call, and that the delay on start-up time can be further reduced if the caller sends initial packets through both relay and direct paths simultaneously before VIA makes the decision. On the other hand, handling a large number call connections at one logical controller presents a scalability challenge, and many existing techniques (e.g., [23]) can be applied.

8 Conclusion

By some estimates, the call volume of Internet telephony surpasses that of traditional telephony. Given its importance, we take the first step towards quantifying the impact of network performance on user-perceived call quality using traces from one of the largest deployed VoIP services. Our sampled dataset consists of 88 million calls over two months. To mitigate calls with poor quality, we revisit the classical overlay network techniques but using the *managed* networks of large cloud providers. Calls between users with poor network conditions can be *selectively relayed* via the managed network. We believe that such managed overlays do not suffer from the drawbacks of traditional overlays. We present VIA which can be viewed as an instance of a managed overlay that carefully picks calls to relay using predicted performance derived from end-to-end measurements collected by the clients, all while dealing with variances in real-world estimates and keeping the volume of relayed calls within a budget. Data-driven evaluation shows that VIA improves call quality by 60% which closely matches the potential indicated by an oracle.

Acknowledgments

We thank Ethan Katz-Bassett for shepherding the paper and the SIGCOMM reviewers for their feedback. We thank Ross Cutler, Jayant Gupchup, Mu Han, Sam Johnson, Tin Qian, Christoffer Rødbro and Ürün Dogan from Microsoft and Skype, whose help made this work possible. We also thank Nimantha Baranasuriya (NUS), Andy Slowey (Microsoft), Amey Karkare (IITK), Aurojit Panda (UC Berkeley), Raajay Viswanathan (UW Madison), Sivaramakrishnan Satyamangalam Ramanathan (USC), Yi Sun (ICT China), Vinayak Naik (IIITD), Akshay Srirangam Narashiman (Delft), Vivek Yenamandra (Ohio), Tarun Mangla (Gatech), Shibshankar Das (IISC), Byung-Gon Chun (Seoul), Bertil Jeppsson, and Javier Aibar Romero from Skype, who have contributed machines and help towards our real-world experiment. Junchen Jiang had an affiliation with Microsoft during the course of this work and at CMU he was supported in part by NSF award CNS-1345305 and Juniper Networks Fellowship.

9 References

[1] G.107: The E-Model, a computational model for use in transmission planning. https://www.itu.int/rec/T-REC-G.107-201506-I/en.

[2] G.114: ITU Recommendation of One-way Transmission Time. https://www.itu.int/rec/T-REC-G.114/en.

[3] Microsoft: Skype runs on Windows Azure; SkyDrive up next. http://www.zdnet.com/article/microsoft-skype-runs-on-windows-azure-skydrive-up-next/.

[4] Quality of Service for Voice over IP. http://www.cisco.com/c/en/us/td/docs/ios/solutions_docs/qos_solutions/QoSVoIP/QoSVoIP.pdf.

[5] Skype's Incredible Rise, in One Image. http://blogs.wsj.com/digits/2014/01/15/skypes-incredible-rise-in-one-image/.

[6] S. Agarwal and J. R. Lorch. Matchmaking for Online Games and Other Latency-sensitive P2P Systems. In *In SIGCOMM*, 2009.

[7] D. G. Andersen, H. Balakrishnan, M. F. Kaashoek, and R. Morris. Resilient Overlay Networks. In *SOSP*, 2001.

[8] P. Auer, N. Cesa-Bianchi, and P. Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, 47(2–3), 2002.

[9] S. Banerji, B. Bhattacharjee, and C. Kommareddy. Scalable Application Layer Multicast. In *SIGCOMM*, 2002.

[10] W. Cao, J. Li, Y. Tao, and Z. Li. On top-k selection in multi-armed bandits and hidden bipartite graphs. In *Advances in Neural Information Processing Systems*, pages 1036–1044, 2015.

[11] R. Castro, M. Coates, G. Liang, R. Nowak, and B. Yu. Network Tomography: Recent Developments. *Statistical Science*, 19(3):499–517, 2004.

[12] C.-N. Chen, C.-Y. Chu, S.-L. Yeh, H. hua Chu, and P. Huang. Measuring the Perceptual Quality of Skype Sources. In *ACM SIGCOMM Workshop on Measurements up the Stack (W-MUST)*, 2012.

[13] C.-N. Chen, C.-Y. Chu, S.-L. Yeh, H. hua Chu, and P. Huang. Modeling the QoE of Rate Changes in SKYPE/SILK VoIP Calls. In *ACM Multimedia*, 2012.

[14] K.-T. Chen, C.-Y. Huang, P. Huang, and C.-L. Lei. Quantifying Skype User Satisfaction. In *SIGCOMM*, 2006.

[15] M. Chesire, A. Wolman, G. M. Voelker, and H. M. Levy. Measurement and Analysis of a Streaming Media Workload. In *Usenix USITS*, 2001.

[16] R. G. Cole and J. H. Rosenbluth. Voice over IP Performance Monitoring. *ACM SIGCOMM Computer Communication Review*, 31(2):9–24, 2001.

[17] F. Dabek, R. Cox, F. Kaashoek, and R. Morris. Vivaldi: A Decentralized Network Coordinate System. In *SIGCOMM*, 2004.

[18] F. Dobrian, V. Sekar, A. Awan, I. Stoica, D. Joseph, A. Ganjam, J. Zhan, and H. Zhang. Understanding the Impact of Video Quality on User Engagement. In *SIGCOMM*, 2011.

[19] F. Dobrian, V. Sekar, A. Awan, I. Stoica, D. A. Joseph, A. Ganjam, J. Zhan, and H. Zhang. Understanding the impact of video quality on user engagement. In *Proc. SIGCOMM*, 2011.

[20] H. Eriksson. MBone: The Multicast Backbone. *Communications of the ACM (CACM)*, Aug. 1994.

[21] P. Francis, S. Jamin, C. Jin, Y. Jin, D. Raz, Y. Shavitt, and L. Zhang. IDMaps: A Global Internet Host Distance Estimation Service. *IEEE/ACM Trans. Netw.*, 9(5):525–540, Oct. 2001.

[22] R. Frederick, V. Jacobson, and P. Design. RTP: A Transport Protocol for Real-time Applications. *IETF RFC3550*, 2003.

[23] A. Ganjam, F. Siddiqi, J. Zhan, I. Stoica, J. Jiang, V. Sekar, and H. Zhang. C3: Internet-scale control plane for video quality optimization. In *To appear in NSDI*. USENIX, 2015.

[24] O. Haq and F. R. Dogar. Leveraging the Power of Cloud for Reliable Wide Area Communication. In *ACM Workshop on Hot Topics in Networks*, 2015.

[25] R. Kateja, N. Baranasuriya, V. Navda, and V. N. Padmanabhan. DiversiFi: Robust Multi-Link Interactive Streaming. In *ACM CoNext*, 2015.

[26] W. Kho, S. A. Baset, and H. Schulzrinne. Skype Relay Calls: Measurements and Experiments. In *IEEE Infocom Global Internet Workshop*, 2008.

[27] H. V. Madhyastha, T. Isdal, M. Piatek, C. Dixon, T. Anderson, A. Krishnamurthy, and A. Venkataramani. iplane: An information plane for distributed services. In *USENIX OSDI '06*.

[28] T. E. Ng and H. Zhang. Predicting Internet Network Distance with Coordinates-based Approaches. In *IEEE INFOCOM*, 2002.

[29] V. N. Padmanabhan and L. Qiu. The Content and Access Dynamics of a Busy Web Site: Findings and Implications. In *SIGCOMM*, 2000.

[30] D. Pendarakis, S. Shi, D. Verma, and M. Waldvogel. ALMI: An Application Level Multicast Infrastructure. In *Usenix USITS*, 2001.

[31] S. Saroiu, K. P. Gummadi, R. J. Dunn, S. D. Gribble, and H. M. Levy. An Analysis of Internet Content Delivery Systems. In *OSDI*, 2002.

[32] S. Savage, A. Collins, E. Hoffman, J. Snell, and T. Anderson. The End-to-end Effects of Internet Path Selection. In

SIGCOMM, 1999.
[33] R. Torres, A. Finamore, J. R. Kim, M. Mellia, M. M. Munafo, and S. Rao. Dissecting Video Server Selection Strategies in the YouTube CDN. In *ICDCS*, 2011.
[34] P. Wendell, J. W. Jiang, M. J. Freedman, and J. Rexford. DONAR: Decentralized Server Selection for Cloud Services. In *SIGCOMM*, 2010.
[35] H. Xie and Y. R. Yang. A Measurement-based Study of the Skype Peer-to-Peer VoIP Performance. In *IPTPS*, 2012.
[36] Y. Xu, C. Yu, J. Li, and Y. Liu. Video Telephony for End-consumers: Measurement Study of Google+, iChat, and Skype. In *IMC*, 2012.
[37] Y. Zhang, M. Roughan, N. Duffield, and A. Greenberg. Fast accurate computation of large-scale ip traffic matrices from link loads. In *ACM SIGMETRICS Performance Evaluation Review*, volume 31, pages 206–217. ACM, 2003.