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 Internet for voice calls is here to stay. In spite of the volume and importance of Internet telephony, we have little understanding of (1) how network performance impacts user-perceived 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 across 1900 ASes and 126 countries. We observe that call quality problems are quite pervasive. More importantly, these problems are *significantly spread out* geographically and over time, thereby making simple fixes targeted at specific "pockets" of poor performance largely ineffective.

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 the emergence of private backbones in recent years to connect globally distributed datacenters, which can serve as a readily available infrastructure for a *managed* overlay network. Trace-driven analysis shows that an oracle-based overlay can *potentially* improve up to 53% of calls whose quality is impacted by poor network performance. A key challenge is realizing these benefits in practice, in the face of significant spatial and temporal variability in performance and 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. Trace-driven analysis and a small-scale deployment shows that VIA cuts the incidence of poor network conditions for calls by 45% (and for some countries and ASes by over 80%) while staying within a budget for relaying traffic through the managed network.

SIGCOMM '16, August 22-26, 2016, Florianopolis , Brazil

DOI: http://dx.doi.org/10.1145/2934872.2934907

CCS Concepts

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

Keywords

Internet telephony; Quality of experience; 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, 6]. The importance of audio calling is evident as almost all major content and social networking platforms today offer 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 (e.g., Netflix) is *interactivity*. The real-time interactive nature of audio calls make them far more sensitive to issues such as high RTT, packet loss and jitter induced by the network [15] while at the same time making solutions based on application-level buffering, that work well with video streaming, less effective.

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 for Internet telephony, and (2) the typical characteristics of performance issues in the real world.

As a first step to bridge this disconnect, we analyze measured network performance and user-perceived quality indexes from one of the largest deployed VoIP services, Skype, which serves hundreds of millions of users and handles a total call volume in excess of a billion minutes of talk-time per day. Our dataset consists of a sample of this total call volume and includes 430 million calls from seven months spanning 135 million users 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 poor networks are not confined to a few bad 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 classical idea of overlay routing (e.g., [8, 33]), which has seen well over a decade's worth of research. Overlay networks can improve performance 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 deploy 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 [25, 37].

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. For instance, Microsoft could choose to route (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 scale.

We evaluate the potential of such relaying using trace-driven analysis of the Skype dataset using their production relays. We find 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 network performance through various relaying alternatives requires carefully tracking these dynamics. Also, operators may prefer to not overload their managed backbone, thus requiring our solution to function within a "budget" for how much traffic could be relayed.

VIA uses an approach called *prediction-guided exploration* to decide which calls to relay and to pick the relay(s). It makes these decisions with 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*.

Specifically, 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 seeks to explore all possible choices, and prediction-based approaches which attempt to predict a single best choice from history information. In addition, VIA also uses network tomography to expand the coverage of prediction to paths that have not even been seen.

Trace-driven simulations show 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 the impact of network performance on Internet telephony audio call quality at scale. (§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

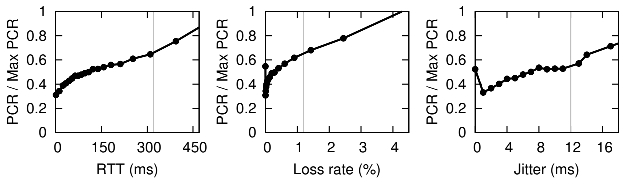
In this section, we use call logs from Skype (described in §2.1) to quantify the impact of network metrics on audio call quality (§2.2), and the patterns of poor network performance (§2.3 and §2.4). opportunities for improving network performance and show why simple fixes are not sufficient. These observations motivate 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 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.,

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

Table 1: Skype dataset summary.



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

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

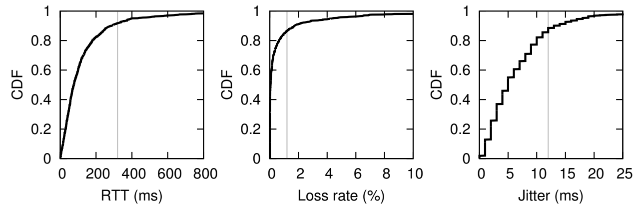
firewall or NAT traversal) rather than for performance optimization (which is our focus here). That is, the only instances of relaying in our passively collected dataset correspond to the caller and callee being unable to establish a direct connection. Despite this bias, the dataset offers a *panoramic* view across diverse end-points from $1,905$ ASes across 126 countries. Table 1 summarizes the 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 several characteristics that make this dataset stand out: a large fraction of calls are international (46.6%), inter-AS (80.7%), and wireless (83%). These characteristics 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., [7]) or confined to server-client paths (e.g., [20]) or academic sites (e.g., PlanetLab [28]).

Each call is associated with three metrics of network performance: (i) round-trip time (RTT), (ii) loss rate, and (iii) jitter. (We do not analyze bandwidth given the low data rate typical of VoIP streams.) These network metrics are calculated by the Skype clients in accordance with the RTP specifications [23] 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.) 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). 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., [17]).



(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.

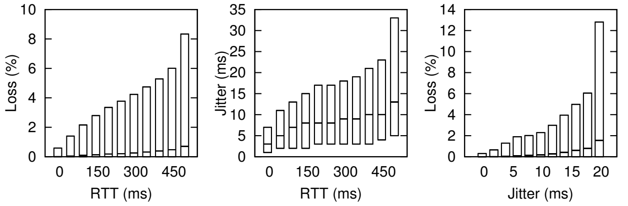
In this section, we show that both PCR and MOS are well-correlated with network metrics. Then, we identify suitable *thresholds* for poor call performance on the network metrics of RTT, loss and jitter. Since our goal is to understand the impact of network performance metrics on call quality, the thresholds keep our focus directly on these network 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 suggests that any improvement in RTT, loss or jitter is likely to improve PCR. MOS (calculated using the model in [17]) also drops with increase in all three metrics (not shown).

Thresholds of network performance: Figure 2 shows the distribution of network performance experienced by calls using default routes. 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 (e.g., loss burst) in bad performance.

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

However, as there could be dependencies between network metrics, improving one metric may increase PNR of another metric. Figure 3 shows the three pair-wise corre-



(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.

lations. 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., minimizing how often *at least one* of the metrics is poor.

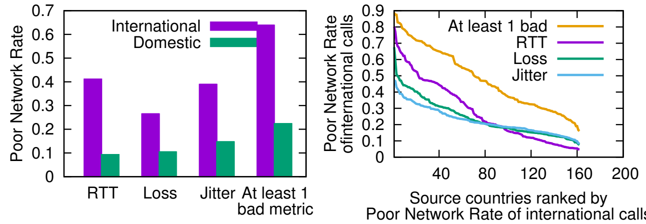
How well does PNR on average values compare to using full packet traces? Analysis of a subset of (70K) calls with full packet traces shows that 80% of calls rated "non-poor" using the thresholds on average metrics ("at least one poor metric") have a (packet-trace based) MOS score higher than three-quarters (75%) of calls rated "poor" using the average metrics. We run a proprietary MOS calculator on the packet traces that contain send/receive timestamps for each packet and loss information. This shows that defining the thresholds on average values of the call is a reasonable approximation.

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., [26]) 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. However, in cases of a poor last-hop network, no relaying strategy can help improve call quality (see §3).

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 suffer from poor performance when using 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 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



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

Figure 4: International vs. Domestic Calls.

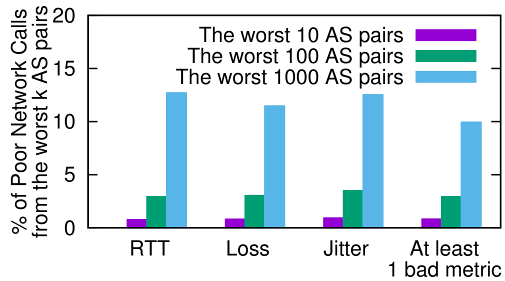


Figure 5: The percentage of calls over poor network conditions that come from the worst n AS pairs; AS-pairs are ranked in descending order of their contribution to total amount of calls with poor performance.

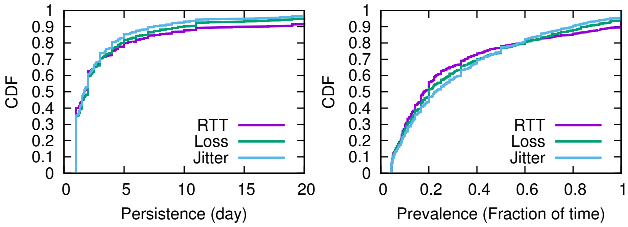
pair of bars), where the gap between international and domestic calls is even larger. Though conclusively diagnosing the root cause of bad performance on international calls is hard and beyond the scope of this work, the higher PNR for international calls points to the WAN path as the culprit.[1]

To understand this further, 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* account for a big chunk of the PNR. Figure 5 shows the fraction of calls that suffer from poor network performance 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

[1] One aspect is that users tend to use VoIP regardless of its performance for international calls, unlike domestic 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.

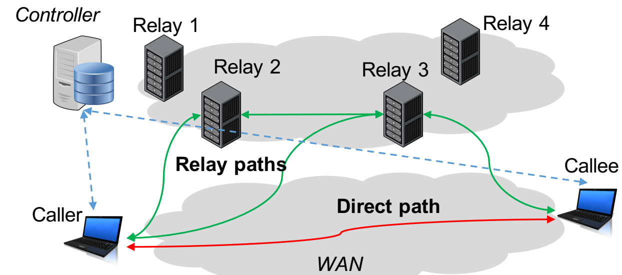


Figure 7: VIA architecture with relay nodes at globally distributed data centers. A call can either take "default path" (red) or a "relay path" (green).

PNR. This means that localized 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, 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 into 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. This observation 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

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* (across ASes) and *temporally*.

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).

3 Approach and Potential of VIA

In this section, we present VIA, a managed overlay architecture that consists of relays hosted at globally distributed data centers and a centralized controller dynamically selecting relays for audio calls (§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 oracle-based scheme for 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%.

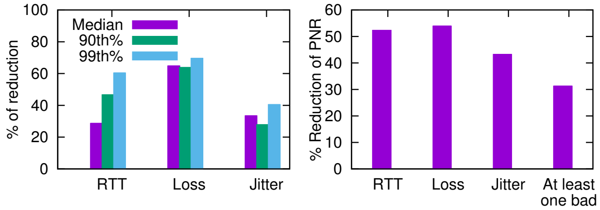
3.1 VIA Architecture

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 [37], 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 relay(s) they should use. The controller makes this decision based on the performance measurements from historical calls and policy



(a) Performance distribution (b) Poor Network Rate

Figure 8: Potential improvement of VIA.

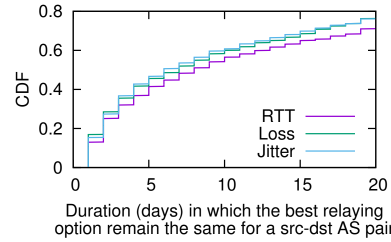


Figure 9: Distribution of how long the best relaying option (picked by oracle) lasts. The optimal relaying options for 30% of AS pairs last for less than 2 days.

constraints (such as those based on relay budget or current load), to be described in §4. To aid in this process, Skype clients periodically 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. (We discuss implementation issues in §7).

3.2 Potential relaying improvement

Next, 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 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— either a relay path or the direct path.[2] 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 indicative samples of its actual performance. In §4.6, we will relax the first assumption by introducing a budget constraint on the 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.

The oracle can help reduce RTT, loss and jitter by 30%-60% at median (Figure 8a). Reduction at the tail, which is of particular significance in interactive services, 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, left three bars) of up to 53%. Source-destination pairs with fewer calls between them have a lower impact on the PNR and its improvement.

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 $\leq$ 320ms, loss $\leq 1.2\%$, *and* jitter $\leq$ 12ms), while still optimizing for RTT, loss and jitter individually. Even while optimizing for each of the three metrics, we can obtain a PNR for "at least one bad" metric; we conservatively pick the *worst* among the three for our analysis. Despite this strict stipulation, we can achieve reduction of over 30% in PNR (Figure 8b, right-most bar).

Need for dynamic relay selection: Whether the controller should select relay dynamically 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. 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 relaying through VIA could provide significant gains, we now devise a *practical* algorithm for relay selection. We begin by formulating the problem of relay selection. We describe two classes of strawman approaches — purely *predictive* and *exploration-based* — and highlight limitations of both classes. We then present the core intuition behind our relay selection algorithm, called *prediction-guided exploration* and then describe the solution.

4.1 Problem formulation

Our goal is to assign each call to a particular relaying option as discussed in §3.1. Recall that a relaying option can use the default path, use a specific one-hop relay node (i.e., bouncing relaying), or use a specific pair of relay nodes (i.e., transit relaying). Let C denote the set of calls we want to op-

[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.

timize 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; i.e., 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 Assign : $C \rightarrow R$ denote the assignment function output by some algorithm and let 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

We can consider two classes of approaches for the optimal assignment of relaying options to 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 classes of approaches individually 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 $O(N^2 M)$ choices. Furthermore, certain end-points 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.

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

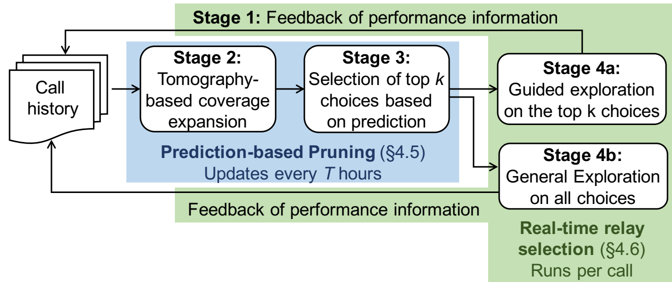


Figure 10: Overview of VIA relay selection based on prediction-guided exploration.

4.3 Overview of our approach

The key intuition behind our solution is the empirical observation that even though a prediction-based approach may not predict the optimal choice, the optimal is likely in the top few of its predictions. In other words, if we look at the top-k choices (those who have the best predicted performances), the optimal choice will likely be a member of that set.

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 R of all possible strategies, 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 options (e.g., [11]). 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.

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
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. 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.

4.4 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: direct paths, and each transit and bouncing relay.

Input: Set of calls C to be assigned to relaying options R, and set of historical calls H
Output: A relay assignment, Assign, where each call $c \in C$ is assigned a relay option Assign$(c) \in R$

```
   /* Stage 2: Tomography-based performance
      predictor trained from H                          */
1  Pred ← BuildPredictor(H)
   /* Stage 3: Pick Top-k candidates based on
      history-based prediction.                         */
2  Assign ← ∅
3  for (s, d) do
4      TopK← GetTopK(s, d, R, Pred)
       /* See Algorithm 2                               */
5      for c ∈ C do
6          if RandomFloat(0, 1) < ε then
               /* Stage 4a: Explore the Top-k
                  candidates                            */
7              r ← Explore(c, s, d, TopK, Assign, Pred)
8          else
               /* Stage 4b: Randomly explore all
                  relaying options                      */
9              Assign(c) ← Random(R)
10         Assign(c) ← r
11 return Assign
```

Algorithm 1: Relay selection algorithm of VIA

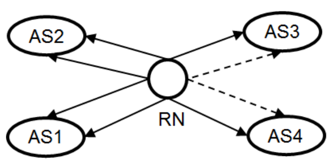


Figure 11: Path stitching in VIA to estimate performance through relay RN. Solid lines represent historical call samples that we use to predict performance between AS3 and AS4 (dotted line). $\mathbf{RTT_{AS3\leftrightarrow AS4} = RTT_{AS1\leftrightarrow AS4} + RTT_{AS2\leftrightarrow AS3} - RTT_{AS1\leftrightarrow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

Expanding coverage by network tomography: Call history tells us the performance of 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 Skype were only designed to forward traffic and we were not in a position to add new functionality to these relay nodes (and potentially impose additional overheads).

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 [12]).

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

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 $60\% - 80\%$ 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 ($\text{Pred}_{lower}(s, d, r)$) of any relay option not in the top-k is higher than the upper 95% confidence bound ($\text{Pred}_{upper}(s, d, r)$) 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%.

4.5 Exploration-exploitation step

Exploring the top-k choices for each source-destination pair (Explore of line 10 in Algorithm 1) can be formulated as an instance of 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.

Relay options selected by the basic *exploration-exploitation*

Input: Source AS s, destination AS d, relaying options R, and predictor Pred (from Algorithm 1)
Output: Top k relaying options TopK for (s, d) calls

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

Algorithm 2: Predicting the top-k choices.

process assigns a fraction of calls to explore different relay options (ϵ-greedy) and the rest to exploit the best decision.[3] As briefly mentioned earlier, standard exploratory approaches are slow to converge (§4.2) and often fail to select the best decision (§5.3). This is because exploring in presence of high variability requires a lot of samples, which is infeasible due to data sparseness and skew.

Algorithm 3 shows the pseudocode of our approach. Here, we choose the UCB1 algorithm [9] as our basic starting point. 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 the basic algorithm in order 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 (e.g., unusually large RTT). Normalizing all values based on such a wide range leads to poor decisions because 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 §4.4 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.

[3]Exploration-exploitation could also be invoked on per-packet basis within the call. However, this would require packet-level control, which is out of the scope of this paper.

Input: Call c, source AS s, destination AS d, top k relaying options TopK, relay assignment Assign, predictor Pred
Output: Relay selection of c

Function $\mathsf{Explore}(c, s, d, TopK, \mathsf{Assign}, \mathsf{Pred})$
/* Initializing variables */
$\text{ucb}_{\text{min}} \leftarrow \infty$; $r_{\text{top}} \leftarrow$ null
/* To avoid outliers, we do not use maximum performance as normalizer w. */
$w \leftarrow \frac{1}{|\text{TopK}|} \sum_{r \in \text{TopK}} (\mathsf{Pred}_{\text{upper}}(s, d, r))$
/* 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. */
$\text{ucb} \leftarrow \frac{1}{w|C_r|} \sum_{c' \in C_r} Q(c', r) - \sqrt{\frac{0.1 \log T}{|C_r|}}$
if $ucb < ucb_{min}$ **then**
$r_{\text{top}} \leftarrow r$
$\text{ucb}_{\text{min}} \leftarrow \text{ucb}$
return r_{top}

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

4.6 Budgeted relaying

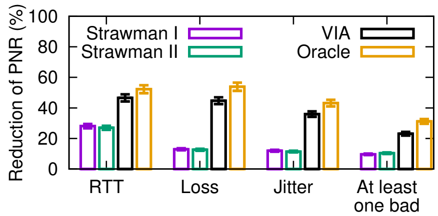
We extend VIA's relaying decision to consider budget constraints: so the fraction of calls being relayed must be less than a certain limit, B (e.g., 30%). While such an overall budget on relayed calls is simple, in general it may also be of interest to consider other budget models, such as per-relay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

VIA utilizes the budget using a simple extension to the heuristic in §4.5. 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.

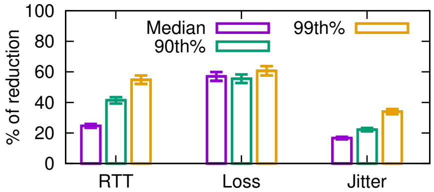
5 Evaluation

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 (compared to the oracle's 30%-60%; §3) and $35\% - 60\%$ on 99^{th} percentile. VIA reduces PNR by $39\% - 45\%$ for the individual metrics (compared to the oracle's 53% in Figure 8b), and by 23% when PNR is computed on an "at least one bad" metric (compared to the oracle's 30%).
- VIA achieves close-to-optimal performance under budget constraints by selectively relaying calls that have higher potential benefit (§4.6).
- VIA's improvement increases as relay decisions are made at finer spatial granularities and more dynamically. However, we start to see diminishing gains at granularities finer than AS-pair and daily.

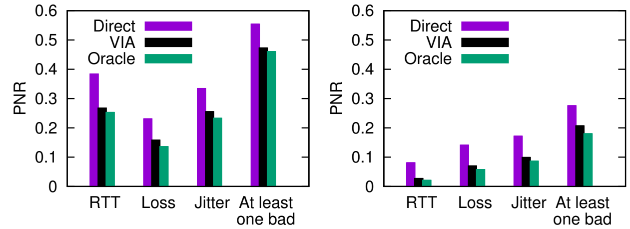


(a) VIA, strawmen, oracle vs. default



(b) VIA improvement on percentiles

Figure 12: Improvement of VIA. PNR on individual metrics improve by $39\% - 45\%$ **and on the "at least one bad" metric by** 23%**.**



(a) International (b) Domestic

Figure 13: VIA improvement on international and domestic calls. We also have similar observation regarding inter-domain and intra-domain calls.

5.1 Methodology

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 (using newer call measurements). 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 in the same 24-hour window. Tomography-based performance prediction is made based on call performance in the last 24-hour window. For statistical confidence, in each 24-hour window, we focus on AS pairs where there are at least 10 calls on at least 5 relay options [4]. Also, the relaying options considered for a call are only those with at least 10 call samples. 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.

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 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 and pure exploratory strategies (§4.2).

Improvement on percentiles: Figure 12b shows the improvement over default strategy on different percentiles. 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 90^{th} percentile), which shows VIA is able to improve the performance of a wide spectrum of calls.

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.

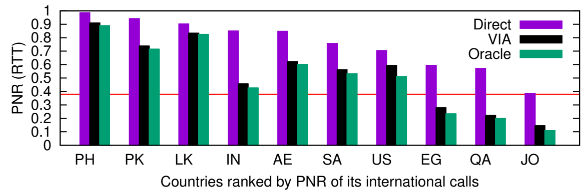
International vs. domestic: Figure 13 compares PNR of international and domestic calls under strategies of default, VIA and oracle.We see significant improvement of VIA on both international and domestic calls, while international calls have a slightly higher magnitude of improvement than domestic calls. 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 14 further dissects the improvement of VIA by countries (with one side of the international call in that country) with worst (direct) PNR. It shows that the worst countries have a much higher (direct) PNR than the global PNR, shown by the horizontal red line, and that the performance of VIA is closer to the oracle than to the default for most of these countries.

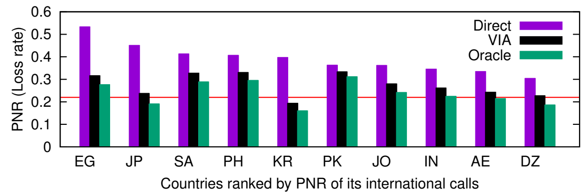
5.3 VIA's design choices

Prediction accuracy of relay-based tomography: As a

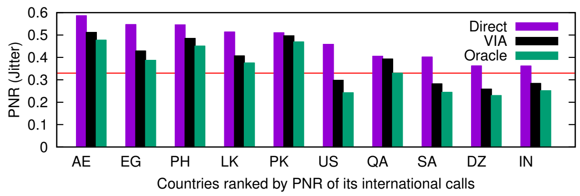
[4] Otherwise, selecting relays from a handful of candidates would be trivial. 32 million calls remain after these filters.



(a) PNR of RTT



(b) PNR of loss rate



(c) PNR of jitter

Figure 14: Dissecting VIA improvement on PNR by country of one side. There is a substantial diversity on VIA improvement across different countries.

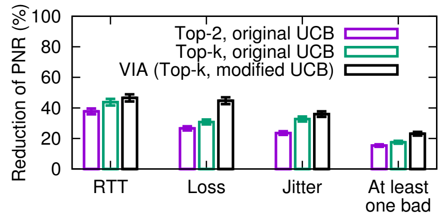


Figure 15: Comparing guided-exploration strategies.

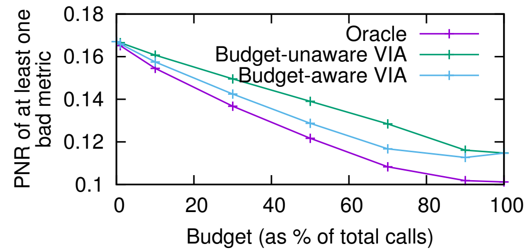


Figure 16: Impact of budget constraint on VIA.

first step, VIA uses relay-based tomography (§4.4) 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 Strawman I (pure prediction-based) that we have seen in Figure 12a, and also motivates real-time exploration.

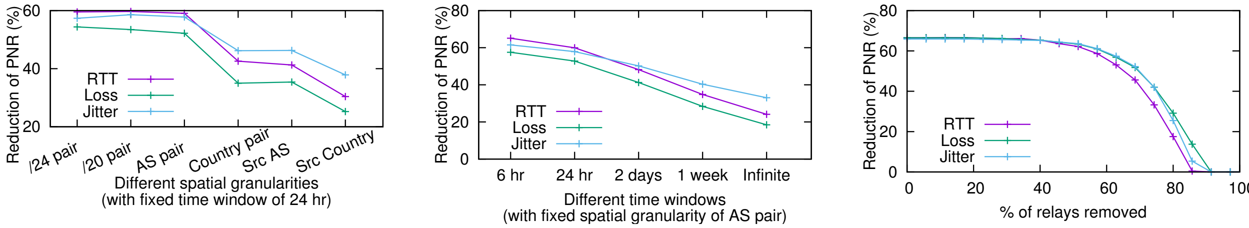
Benefits of prediction-guided exploration: As discussed in §4, VIA is not a simple combination of prediction and exploration approach. First, instead of picking a fixed number top candidates, VIA pick top candidates by taking variance of prediction into account. Second, instead of using the original UCB1 algorithm, which assumes a normal distribution of rewards, we adopt a different way to normalize values to cope with performance outliers. Figure 15 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 reduces PNR by 24% compared to 15% with just the top 2 (loss rate PNR by 44% compared to 26%).

5.4 Practical relaying factors

Relaying budget: 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 (§4.6) 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 (§4.6) 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).

Relaying 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 (Figure 10) 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. Second, 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



(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.

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 (§4.3).

Relay usage: 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. Removing 50% of the (least used) relays causes little drop in VIA's gains. This suggests that new relays should be deployed carefully in future.

5.5 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 clients. In total, it created 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. For simplicity, we omit the direct path as an option.

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 $\frac{\mathrm{Perf}_{\mathrm{VIA}}-\mathrm{Perf}_{\mathrm{oracle}}}{\mathrm{Perf}_{\mathrm{oracle}}}$. We found that VIA's relaying decision is within 20% of an oracle's performance for 70% of the calls. Note that this is despite picking the *best* relay (i.e., sub-optimality of 0) for no more than 30% of the calls. When there are multiple relaying options with similar performance, temporal fluctuations may lead to not always picking the best option. But VIA usually picks the option that is close in performance to the best.

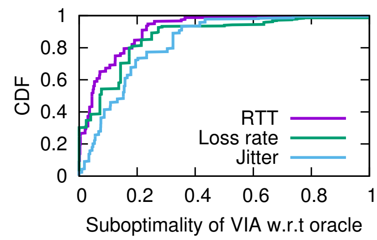


Figure 18: Deployment results. CDF, over calls, of sub-optimality (lower is better) of VIA's performance.

6 Related Work

Overlay routing: Overlay networking has been explored in a variety of contexts, such as virtual private networks (VPNs) and multicast [21, 31, 10]. Of interest to us here is work focused on overlay routing with a view to improving performance [33, 8]. 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 waypoints.

Despite this promise, 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.

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 [27]. However, in the recent years, Skype has moved to a hybrid model [37], 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 [37].

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 content distribution networks [35, 34]. 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 [22, 29, 28], 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 [18] or estimate the performance of network segments using network tomography techniques [12]. 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 [30], CDNs [32], and video-on-demand streaming [16, 19]. There have also been studies of audio-video conferencing by working outside the system, say by running active measurements to Skype super-nodes [36] or sniffing traffic in modest-size deployments [37]. To our knowledge, this work 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 [17, 15, 36, 13, 14]. These models vary in the particular network metrics and codecs they consider. In Section 2.2, we used the model proposed in [17], which is based on the E-Model defined by the ITU [1].

7 Discussion

Cost of centralized control in VIA. Our pilot deployment and client modifications suggest a feasible path to a large-scale deployment from a software update and engineering perspective. One potential concern, however, is the scalability and responsiveness of the control platform. On the one hand, VIA introduces minimal per call overhead, since the client-controller communication need only consist of one measurement update and one control message exchange per call and can be further reduced if the clients cache the best relaying options. On the other hand, handling a large number of call connections at one logical controller presents a scalability challenge, though partitioning techniques provide a good starting point. Also, we conjecture that approaches similar to the split-control architecture employed in C3 [24] might offer a scalable realization, since the measurement and control exchange of the C3 controller (which directs clients to video CDNs) is similar to the measurement and control needed for a large-scale VOIP relay server.

Hybrid reactive decentralized approaches. A natural alternative to relay selection is to simply have clients try a list of relay options sequentially or in parallel, and pick the best option. Such an approach may be good enough for long-lived calls. This would avoid the overhead of data collection and generating the network map. However, as we discussed earlier, this may not be feasible given the large search space of relaying options. An interesting hybrid approach is using the prediction-guided exploration observations as a means to *prioritize or prune* this approach. We intend to explore this approach going forward.

Active Measurements. While our current solution relied entirely on passive measurements from client calls, there is an opportunity to augment it with *active* measurements (by making mock calls between users or from users to relays), especially since the client software can be readily controlled to make them. Active measurements can be intelligently orchestrated to fill “holes” in the passively obtained measurements, thereby making our prediction-guided exploration (both its aspects—tomography as well as bandit solution) more effective. Doing so will require considering the additional load imposed on the clients due to the collection.

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 call quality using traces from Skype, one of the largest VoIP services. Our sampled dataset consists of 430 million calls over seven 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. Such managed overlays do not suffer from the drawbacks of traditional overlays.

To leverage such a managed overlay infrastructure, we present the design of VIA, a system that carefully selects a subset of calls to be relayed using the managed overlay. VIA uses a guided exploration procedure using predicted performance derived from end-to-end measurements collected by the clients, 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 45% which closely matches the potential benefits indicated by an oracle.

Acknowledgments

We appreciate the feedback by the anonymous SIGCOMM reviewers, and thank Ethan Katz-Bassett for the caring and diligent shepherding of the paper. This work would not have been possible without the help of Ross Cutler, Jayant Gupchup, Mu Han, Sam Johnson, Tin Qian, Christoffer Rødbro, Ürün Dogan, and Andy Slowey, Bertil Jeppsson, Javier Aibar Romero from Microsoft. We also thank Nimantha Baranasuriya, Amey Karkare, Aurojit Panda, Raajay Viswanathan, Sivaramakrishnan Ramanathan, Yi Sun, Vinayak Naik, Akshay Srirangam Narashiman, Vivek Yenamandra, Tarun Mangla, Shibshankar Das, Byung-Gon Chun who contributed machines towards our real-world deployment. 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 a 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] WhatsApp Calling: 100 million conversations every day. https://blog.whatsapp.com/10000625/WhatsApp-Calling-100-million-conversations-every-day,, Jun 23, 2016.

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

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

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

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

[11] 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.

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

[13] 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.

[14] 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.

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

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

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

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

[19] 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.

[20] 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.

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

[22] 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.

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

[24] 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 *NSDI*. USENIX, 2015.

[25] 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.

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

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

[28] 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*.

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

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

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

[32] 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.

[33] S. Savage, A. Collins, E. Hoffman, J. Snell, and T. Anderson. The End-to-end Effects of Internet Path Selection. In *SIGCOMM*, 1999.

[34] 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.

[35] P. Wendell, J. W. Jiang, M. J. Freedman, and J. Rexford. DONAR: Decentralized Server Selection for Cloud Services. In *SIGCOMM*, 2010.

[36] H. Xie and Y. R. Yang. A Measurement-based Study of the Skype Peer-to-Peer VoIP Performance. In *IPTPS*, 2012.

[37] Y. Xu, C. Yu, J. Li, and Y. Liu. Video Telephony for End-consumers: Measurement Study of Google+, iChat, and Skype. In *IMC*, 2012.