Piyush Kumar Sharma*, Shashwat Chaudhary†, Nikhil Hassija†, Mukulika Maity, and Sambuddho Chakravarty

The Road Not Taken: Re-thinking the Feasibility of Voice Calling Over Tor

Abstract: Anonymous VoIP calls over the Internet holds great significance for privacy-conscious users, whistle-blowers and political activists alike. Prior research deems popular anonymization systems like Tor unsuitable for providing the requisite performance guarantees that real-time applications like VoIP need. Their claims are backed by studies that may no longer be valid due to constant advancements in Tor. Moreover, we believe that these studies lacked the requisite diversity and comprehensiveness. Thus, conclusions from these studies, led them to propose novel and tailored solutions. However, no such system is available for immediate use. Additionally, operating such new systems would incur significant costs for recruiting users and volunteered relays, to provide the necessary anonymity guarantees.
It thus becomes an imperative that the exact performance of VoIP over Tor be quantified and analyzed, so that the potential performance bottlenecks can be amended. We thus conducted an extensive empirical study across various in-lab and real world scenarios to shed light on VoIP performance over Tor. In over half a million calls spanning 12 months, across seven countries and covering about 6650 Tor relays, we observed that *Tor supports good voice quality (Perceptual Evaluation of Speech Quality (PESQ) >3 and one-way delay <400 ms) in more than 85% of cases*. Further analysis indicates that in general for most Tor relays, the contentions due to cross-traffic were low enough to support VoIP calls, that are anyways transmitted at low rates (<120 Kbps). Our findings are supported by concordant measurements using `iperf` that show more than the adequate available bandwidth for most cases. Hence, unlike prior efforts, our research reveals that Tor is suitable for supporting anonymous VoIP calls.

Keywords: Anonymous calls, Tor, VoIP, Privacy

Received ; revised ; accepted .

1 Introduction

Voice-over-IP (VoIP) applications that support traffic encryption are popular among users who are concerned about their communication privacy. However, these applications do not safeguard the anonymity for Internet users residing in regimes which may conduct surveillance (*e.g.*, the ability of NSA to intercept conversations is widely known [3, 4, 11]). Moreover, to the best of our knowledge, there does not exist any functional VoIP based system that ensures communication privacy and anonymity (along with real-time communication guarantees), required by privacy conscious Internet users and whistle-blowers alike.

Popular privacy and anonymity preserving systems like *Tor* [8] hide the actual IP address of the communication peers by routing their traffic via a cascade of proxies. Since such systems reroute traffic via circuitous paths, it is a widely held belief that they would incur intolerable delays for real-time applications like VoIP. More importantly, while traditionally VoIP relies on UDP traffic to ensure real-time guarantees, popular systems like Tor are designed to transport TCP traffic. Further, complex cryptographic handshakes, essential to the anonymity guarantees provided by such systems, may exacerbate the impact on performance, making them unsuitable for real-time applications.

Prior efforts on anonymous calling [7, 13, 22, 38] unanimously agree with aforementioned shortcomings of Tor. Some conclude this based on potentially biased results [36], involving relays only in Europe or only conducted a few hundred calls [13], lacking diversity. Others, such as Le Blond *et al.* [22], did not conduct any experiments to measure VoIP performance over Tor. Over-

***Corresponding Author: Piyush Kumar Sharma:** Indraprastha Institute of Information Technology (IIIT) Delhi, India E-mail: piyushs@iiitd.ac.in
Shashwat Chaudhary†: IIIT Delhi, India E-mail: shashwat15091@iiitd.ac.in
Nikhil Hassija†: IIIT Delhi, India E-mail: nikhil15065@iiitd.ac.in
Mukulika Maity: IIIT Delhi, India E-mail: mukulika@iiitd.ac.in
Sambuddho Chakravarty: IIIT Delhi, India E-mail: sambuddho@iiitd.ac.in
† Both the authors have equal contribution.

all, we believe that existing literature does not quantify the actual interplay of network performance attributes (*e.g.*, one-way delay (OWD), available bandwidth, *etc.*) and how it impacts VoIP call quality over Tor. However, the belief that Tor is not competent enough to transport VoIP traffic is still prevalent [13].

Thus several novel VoIP architectures were proposed [7, 13, 22] to tackle the shortcomings. Some of these, like *Phonion* [13] and *Herd* [22], require a new volunteer-run network with millions of active users (like Tor) for providing anonymity guarantees. This requirement can be a major stumbling block. Moreover, in the absence of active users and significant cross-traffic, comparable to that of Tor (that transports over 200 Gbit/s traffic per day [2]), one cannot adjudge future anonymity and performance assurances of these proposals. *Importantly, no such system is currently functional.*

Hence, in the absence of an existing anonymous voice calling system, we chose to determine the root cause(s) of poor performance over Tor. After ameliorating them, one may expect to achieve adequate voice call quality with anonymity guarantees equivalent to that provided by Tor. To that end, we began by conducting a pilot study where we made VoIP calls over the Tor network. We also captured the network performance attributes, in order to eventually identify how they impact voice call quality. Following ITU guidelines and prior proposals [13], we used one-way delay (OWD) and *Perceptual Evaluation of Speech Quality* (PESQ) [35] as metrics to judge VoIP call quality. The PESQ ascribes a value to judge the user-perceived audio quality in an automated manner.

We made 1000 consecutive calls through individual Tor circuits[1], with the callee and caller machines under our control. Contrary to the prevalent notion of poor quality of calls via Tor, we observed good call quality, with average PESQ $\approx$ 3.8 and average OWD $\approx$ 280 ms. Overall, 85% of the calls were acceptable (PESQ >3 and OWD <400 ms) as per ITU [15, 16].

In the absence of substantial evidence of poor quality calls, we went ahead and conducted an extensive longitudinal study involving 0.5 million voice calls over the Tor network, spread across 12 months. These measurements not only involved varied in-lab and real-world scenarios with diverse Tor relays, VoIP applications, caller-callee locations but also a user study. To our surprise, even then more than 85% of voice calls had acceptable perceptual quality.

The PESQ metric varies inversely with distortions in the perceived audio. Packet drop and jitters, which cause such distortions, are an artifact of increased cross-traffic contentions. *A high PESQ score, accompanied with low overall OWD, in the majority of the cases thus indicates low cross-traffic contentions, for VoIP calls. Moreover, VoIP calls which are mostly transmitted at low bit-rates (<120 Kbps), incur less routing costs, and thus may not suffer much distortions.*

Other network performance metrics like available bandwidth also varies with network cross-traffic volume. Thus, concomitant available bandwidth during the call (over 1 Mbps in 90% cases[2]), along with data published by Tor Metrics [2], confirms the reason for obtaining good results.

Overall, this first ever long-term study involving extensive evaluations of calls over Tor, bore some interesting results and insights. We summarize them as follows:

1. In the vast majority of our experiments (>85%), involving voice calls over individual Tor circuits, we observed acceptable call quality with PESQ above 3 and OWD less than 400 ms. This holds for a diverse set of scenarios:
   (a) Caller and callee spread across 7 countries (in three continents).
   (b) Coverage of a total of 6650 Tor relays, 22 times more than previous studies [36].
   (c) Popular VoIP apps such as Telegram and Skype.
   (d) Both caller and (or) callee using Tor circuits to establish calls with one another.
   (e) Different codecs and call duration.
   (f) User study involving humans rating voice calls.
2. Acceptable performance in the vast majority of cases was due to *relatively low contention for VoIP in most Tor relays.*

2 Background and Related Work

In this section, we begin by describing the basic concepts behind VoIP calling and its evaluation metrics. Next, we briefly describe Tor, followed by a discussion on different types of anonymous calling scenarios. Finally, we describe related work in this domain.

2.1 Basics of VoIP

Voice over IP enables real-time voice communication over IP networks. Any VoIP based system comprises

1 Using the standard Tor client utility.

2 The bandwidth was measured by `iperf`.

of two primary channels: one for control and signaling, and the other for transporting the actual encoded voice traffic. Session Initiation Protocol (SIP) [37], is an example of a popular VoIP signaling protocol. It includes functionality like authenticating users, establishing and terminating calls, *etc.* SIP, along with Session Description Protocol (SDP) [12], allows for negotiating the call related parameters like codecs. Once the call is set-up, Realtime Transport Protocol (RTP) [39], a UDP based protocol, solely manages voice traffic. It adds sequence numbers to packets for in-order delivery, and buffers them to minimize the impact of jitter.

2.2 QoS Metrics for Voice Calling

The following metrics are often used for measuring the quality of voice calls.

1. **Perceptual Evaluation of Speech Quality:** PESQ [35] is the ITU specified and standardized metric for voice call quality evaluation. It estimates the user perceived call quality. PESQ computation requires both the source and recorded audio for comparison. The PESQ metric generates an objective score using its own algorithm which is then mapped to a subjective Mean Opinion Score (MOS). It is demonstrated that the score generated by PESQ highly correlates with the MOS score reported by actual users. The metric largely reflects distortions in recorded audio, which in-turn indicates the impact of network losses and jitters.
   It returns values between 0 and 5, 5 being excellent, and 0 being poor and unusable. However, in practice, values between 1 and 4.5 are observed. Following ITU specifications, in our tests, we considered calls with PESQ greater than three as acceptable.
2. **Jitter and Packet Loss:** Jitter represents variations between subsequent packet arrivals. Such variations may arise due to packet reordering and losses. While VoIP users can endure minor losses, jitter may dramatically hamper the perceived call quality. Even non-permissible variations in either of these two metrics can sharply perturb PESQ, which incorporates the impact of both the metrics.
3. **One Way Delay (OWD):** OWD is the time duration between when voice packets are encoded at the sender and when they are successfully decoded at the receiver. According to the ITU specification [15], OWD should ideally be less than 150 ms. Further, OWD below 400 ms is considered permissible for international calls [16]. Hence, we chose the permissible limit of 400 ms to evaluate the performance of voice calls.

Interestingly PESQ score, other than losses, only captures the impacts of jitters, which represents variations in OWD. It would suffer no perturbations if all the voice packets were uniformly delayed, without jitter or losses. However, for humans, such delays lead to time-shifted audio, eventually leading to poor perceptual quality. Thus, in our evaluation study we considered both PESQ and OWD as metrics for estimating call quality.

2.3 The Onion Router (Tor)

Tor [8] is a widely-used low-latency anonymization network. It allows it's users to communicate without revealing their IP addresses. It consists of globally distributed volunteered hosts acting as relays. Clients communicate to servers by proxying their traffic via a cascade of three such relays, *viz.*, the *entry*, the *middle*, and the *exit* nodes. The client encrypts the traffic using a three-layered encryption scheme, each corresponding to the three relays, using keys negotiated with each of them, respectively. These encrypted packets are then forwarded via the three chosen relays. Each of these relays de-crypts one layer of encryption and forwards it to the next one in the cascade. Thus, no one ever, other than the client itself, knows *the IP address of all the relays and the server.* Each relay only knows about the previous and next one in the cascade. The server only sees connections arriving from the exit node, but knows nothing about the client. By design Tor only supports TCP streams.

2.4 Types or Use Case of Anonymous Calling

As already mentioned, anonymous calls are of great use to whistleblowers, activists, undercover reporters, *etc.* However, it may be inquired as to what are the current alternatives (used by such groups) to conduct anonymous calls. To the best of our knowledge, there are no publicly available alternatives. This is the reason that in the past decade, prior efforts [13, 22] attempted to design and build such systems. However, as discussed in Subsec. 2.5, none of these systems are functional. Moreover, there are secure messaging and end-to-end encrypted communication apps such as Signal and Telegram which are highly popular among privacy practitioners. Though these provide security against eavesdroppers, the centralized architecture of all such apps allows the central server to know the details such as

who is calling to whom. Thus, even though such apps provide secure calls, they may not be fit to be used for anonymous calls.

We now describe the anonymous voice calling scenarios tested in our study.

1. **Caller (one-way) anonymity:** Here, the caller wants to achieve anonymity (by hiding IP address) against an adversary that may monitor and (or) filter its traffic. Such scenarios represent journalists and whistle-blowers who communicate sensitive information to other individuals or groups (*e.g.*, news headquarters), while evading the adversary.
2. **Caller-Callee (two-way) anonymity:** Here, both parties want to achieve anonymity. Two individuals who both wish to communicate covertly, while remaining anonymous to their respective adversaries, may use such setups.

Further, these setups and their use cases are discussed in detail in Sec. 4.

2.5 Prior Efforts

There exists scant literature on performance evaluation of voice calls over Tor (ref Fig. 1). The only attempts made were by Rizal *et al.* [36] and Heuser *et al.* [13]. Their efforts involved instantiating a few hundred calls through Tor for evaluating their quality.

However, the aforementioned efforts were limited in scope. *E.g.*, rather than using PESQ (an established call quality metric), Rizal *et al.* relied only on network parameters like OWD, jitter, and packet loss as evaluation criteria. Also, this study involved Tor relays only in Europe (with only 298 out of the available 4453) for their evaluation, and may not be representative of VoIP performance, for the complete Tor network. Based on these preliminary studies, previous researchers ignored Tor and proposed novel architectures [7, 13, 22] for providing anonymous voice calls. Next, we describe all such efforts.

Inspired by Chaum's mixes [6], Pfitzman *et al.* [30] proposed ISDN mixes for anonymous voice communication. Authors proposed two simplex Mix channels, one for the sender and the other for the recipient, thereby enabling full-duplex anonymous communication over the telephony network.

Drac [7] by Danezis *et al.* involves an architecture for anonymous low latency voice communication using social networks as relays to route traffic. The system, while providing anonymity to a particular user, relies on using the identity of other users in its social circle as alibis. However, Drac presents an analytical model with no functional deployment.

Torphone [51], a system designed over Tor, was an extension to send VoIP traffic via Tor. It reported having achieved an OWD of 2–4 s, which is unsuitable for acceptable call quality. However, it is presently non-functional (and was last used on Windows XP).

In 2015, Le Blond *et al.* [22], proposed a novel architecture, *Herd*, to prevent global passive or active adversaries attempts to de-anonymize users, based on call metadata (correlating start and end times of a call). Herd relies on a set of dedicated mixes that relay VoIP traffic to other mixes and endpoints while hiding any distinct traffic patterns. The mixes are also known as *zones*, and a user can select available trustworthy zones to route its call. They believed prior results on evaluating the performance of VoIP over Tor (published back in 2008 [28]), and concluded the RTT to be high (2–4 s), deeming Tor unsuitable for VoIP. However, they refrained from conducting a fresh study to gauge the (then) recent performance of Tor. Additionally, they concluded that the calling entities on Tor could be easily correlated, given the start and end times of a call. They validated this claim by analyzing the call records of a service provider. However, obtaining such data for calls conducted via Tor might not be easy, as Tor is a globally distributed system, and it would require gathering data from geographically diverse ISPs. Moreover, they tested their prototype on only four cloud hosts, performing just 12 calls, in the absence of active users and significant cross-traffic when compared to Tor.

Phonion [13] is one of the recent anonymous VoIP systems. It is fundamentally similar to Tor, but specifically developed for voice communication. It uses *relays* (similar to Tor relay), *relay services* and *broker system* (similar to Tor directory authorities). Phonion attempts to anonymize call data records (CDRs) against different adversaries. In Phonion, the calls are relayed via various service providers. This prevents a single provider to gather all the call records for a particular call. The advantage of Phonion is that it works across different voice calling technologies — VoIP, cellular or PSTN (public switched telephone network). Additionally, it requires Internet access only for an initial bootstrap phase after which, anonymous calls could be instantiated over carrier services. However, the authors of Phonion also side stepped using Tor. Through a pilot study of 100 calls, they concluded Tor to be unsuitable for transporting VoIP calls. The study involved comparing one, two and three-hop circuits of their prototype, with six-hop Tor circuits (two-way anonymity). However, we show

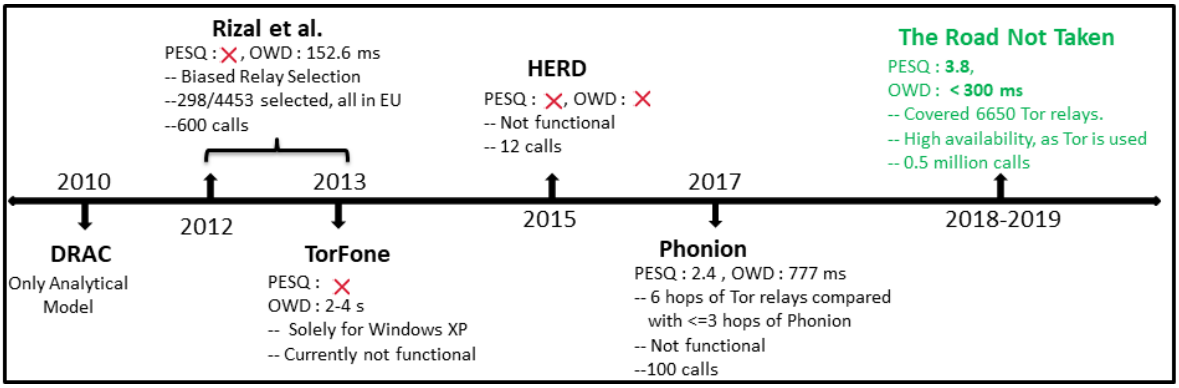

Fig. 1. Existing literature on anonymous voice calling highlighting their inconsistencies and incompleteness. X implies metric not considered or evaluated in the study.

in Sec. 4.2.1 and Sec. 4.2.2 that regular three-hop Tor circuits provide suitable performance for VoIP. Usage of this setup and performing 100 calls might have led them to report an unacceptable average OWD of 777 ms for Tor circuits along with an average PESQ of about 2.4. Moreover, the implementation of their system relied on `google voice`, a service only available in N.America, making the system unusable elsewhere. Lastly, it required the relay operator to pay an operational fee to telcos, which might serve as a stumbling block for recruiting relays.

Overall, a major hindrance in the wide-scale adoption (and availability) of such systems, is the recruitment of new volunteered relays and users (which Tor already has in abundance). Schatz *et al.* [38] also highlight this issue. Additionally, as previously mentioned, prior evaluations over Tor, provide very few insights as to how the interplay of network performance attributes affects the voice call quality.

Comparison with prior VoIP measurements over Tor: As already mentioned, only two studies actually measured the performance of VoIP over Tor — Rizal's study [36] and Phonion [13] by Heuser *et al.* We now explain how our measurement study is methodologically different from them.

Rizal's study: This study transported VoIP traffic over Tor by tunnelling it via VPN tunnels. The experiments conducted in this study, involved Tor relays hosted only in Europe. Thus, Rizal reported a reasonably low average OWD of ≈ 152 ms likely due to the geographic proximity of relays. On the other hand, our study included relays from all parts of the globe and hence provided better coverage of the entire Tor network[3]. Thus, our study measured an average OWD of ≈ 280ms likely due to the diversity of involved Tor relays. Moreover, unlike Rizal's study, we used PESQ as an evaluation metric, which is an industry standard for measuring user perceived voice quality.

Phonion: It utilized the Mumble VoIP software [23] to transport VoIP traffic via Tor. Authors of Phonion used PESQ as an evaluation metric. However, their study involved measuring performance only for two-way anonymity, and not the one-way anonymity. On the other hand, our study involved different scenarios corresponding to both the cases *viz.*, one-way and two-way anonymity. Hence, we involved comparatively more diverse and comprehensive set of measurement setups.

Additionally, both the previous studies involved performing only a few hundred calls — 100 in Phonion and 600 in Rizal. In contrast, we performed ≈ 0.5 million calls involving a diverse set of geographic locations, media codecs, *etc.,* along with controlled experiments involving private Tor setups.

Overall, these studies lack the requisite comprehensiveness and a detailed analysis of different performance attributes that could provide deeper insights for VoIP performance over Tor. Thus, our study aimed to fill this research gap by conducting an extensive study with an intent to better understand the behavior of VoIP calls over Tor in varied setups and network conditions.

Other Tor performance measurement studies : There exist abundant studies which measure the Tor networks' performance in terms of observed bandwidth and latency [9, 17–20, 24, 29, 41, 42, 52]. Few, like Shadow [18] and Chutney [49], developed simulators to analyze the performance of Tor. Others like Torflow [29], EigenSpeed [41], and Peerflow [20] *etc.,* focused on measuring relay bandwidth for calculating relay weights by either directly measuring relay bandwidth (Torflow) or by indirectly inferring it using statistics such as relays

3 Achieved using the standard Tor utility that helped create different circuits based on default Tor circuit selection algorithm.

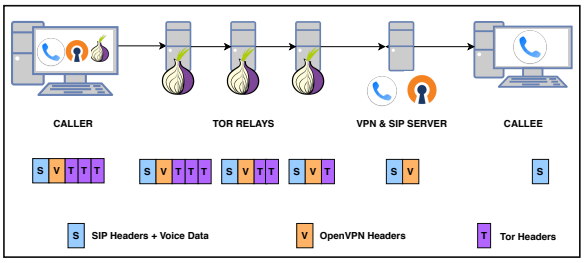

Fig. 2. V-Tor: The caller establishes a VPN tunnel through Tor, so that all the UDP VoIP traffic traverses the Tor network. On reaching the VPN server, the traffic is sent to the SIP server, which initiates a voice call to the callee.

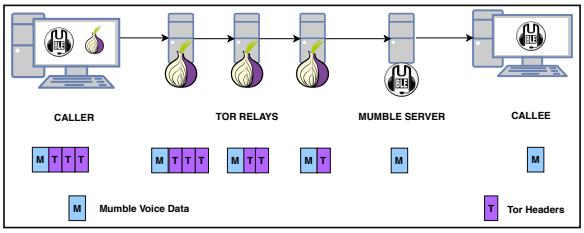

Fig. 3. M-Tor: The caller configures the mumble client to work in TCP mode. Thereafter, mumble's traffic is sent over Tor eventually reaching the mumble server, which handles the call procedure between the caller and callee.

reporting the amount of data exchanged between them (Peerflow). Measuring relay bandwidth is a crucial task in Tor, as it is used to assign weights to different relays that govern the selection probability of a relay in a Tor circuit and thus the amount of traffic the relay might serve. Cangialosi *et al.* [5] specifically focused on measuring RTT between relay nodes. Tor metrics [48] is another popular (and actively maintained) project that periodically measures various performance attributes of the Tor network *e.g.,* circuit RTTs, bandwidth for downloading files of various sizes, *etc.* However, none of these existing projects, measure the perceptual VoIP quality and its associated network characteristics. In our study, we primarily measure the VoIP performance over Tor, along with network characteristics (like bandwidth and RTT) of millions of Tor circuits. The metrics, such as available bandwidth are essential in our study as they help to infer the potential reasons for good or bad performance while conducting VoIP calls. We often referred to the Tor metric results to further support our claims.

3 Measurement Approach

In this section, we describe our experimental setups and the approach taken for performing experiments to measure the quality of voice calls over Tor.

3.1 Experimental Setup

Previous efforts acknowledge that transporting VoIP traffic anonymously over Tor is non-trivial, as VoIP generally uses UDP and Tor only supports TCP. There are two ways in which one can succeed in sending VoIP traffic over Tor. One way is to tunnel UDP packets inside TCP flows. The other way is to directly encode and send VoIP traffic in TCP packets. Thus, some previous studies [36] utilized VPN tunnels to encapsulate and transfer VoIP packets, while others like Phonion [13], relied on Mumble [23] VoIP software to generate TCP packets and transfer them directly to Tor. Similar to these studies, we tested VoIP performance using both the above approaches. These setups are now described in detail below:

1. **SIP client via VPN through Tor *(V-Tor)*:** The V-Tor setup (ref. Fig. 2), involves a SIP client (caller) connecting through a Tor circuit to a VPN server for establishing a TCP tunnel. A step by step walkthrough of this setup is described below:
   Step1 The caller runs the VPN and Tor client utilities. VPN client is configured to establish a VPN tunnel (to the VPN server) over a Tor circuit, by forwarding VPN traffic to the Tor SOCKS interface. This ensured that all the traffic from the caller would reach the VPN server via the Tor network.
   Step2 Client would initiate a VoIP call using a SIP utility. The VoIP traffic (generated from SIP) would reach the VPN server via Tor (explained in step 1). VPN server decapsulates SIP packets and forwards them to the SIP server.
   Step3 SIP server then helps negotiate the call between the caller and the callee.
2. **Mumble with TCP mode over Tor (*M-Tor*):** This setup relies on using the Mumble client program (in TCP mode) for encoding voice traffic through TCP streams. These streams are transported via Tor to a Mumble server that mediates the voice call between the parties.

It must be noted that we performed experiments for both the V-Tor and M-Tor setups. However, since we obtained similar experimental outcomes from both the setups, we present description of V-Tor experiments in the main body of the paper, and the details of M-Tor experiments in Appendix A.

3.2 Overview of Experiments

We now describe the experiments performed to measure the VoIP call quality over Tor. Our experiments were

primarily conducted to identify the root cause of the hitherto believed poor voice call quality of Tor.

We began with a pilot study that involved conducting 1000 consecutive calls over Tor, with the caller and callee under our control, but in different geo-locations. Our results, using V-Tor (and M-Tor) setup, showed high call quality in a large fraction of the cases. Considering these results to be potential outliers, compared to findings of previous authors, we went ahead and conducted a comprehensive measurement study spread across 12 months.

We conducted two different sets of experiments — (1) involving in-lab setups and (2) involving circuits through the public Tor relays.

In-lab experiments: In the in-lab setup, our goal was to measure the performance of VoIP over V-Tor (M-Tor) setups with competing cross-traffic entirely under our control. For this, we setup a private network in our lab, consisting of Tor nodes along with client and server (VPN, SIP, etc.) machines. This private network was deployed on real machines, with three of them serving as relays, while one of them was also serving as a directory authority. Other machines acted as clients and servers. In these experiments, we measured performance attributes and established baseline values (*e.g.,* bandwidth requirement) for a VoIP call. These experiments were performed over setups involving different combinations of VPN, Mumble, and Tor in the following manner:

1. *Direct VoIP calls:* VoIP calls between the caller and callee were conducted without involving Tor and VPN. The caller and callee communicated directly using SIP or Mumble protocol. This helped us in observing the minimum bandwidth and delay requirements when no overhead was introduced (due to Tor or VPN).
2. *VoIP calls over VPN:* Caller and callee communicated using SIP protocol. However, calls were encapsulated through VPN connections, in order to measure the impact (if any) due to the overhead of running a VPN.
3. *VoIP calls over Tor:* Encapsulating the calls through VPN connections (or Mumble) *and* then transporting them via Tor circuits, to evaluate the impact of the overheads due to Tor.

Moreover, we performed some additional experiments to observe the impact of variation in background cross-traffic on VoIP calls. We observed the variation in performance when VoIP call(s) were made in the presence of heterogeneous background traffic (*e.g.,* other VoIP calls and web traffic). Such in-lab experiments may potentially present clues regarding the number of VoIP clients that could be simultaneously supported by the real Tor network.

Internet based experiments over public Tor: After performing various in-lab tests, we carried out multiple experiments involving public Tor relays, where we had no control over the background cross-traffic and network conditions. These experiments involved measuring performance across diverse scenarios (Tor relays, endpoints, codecs, *etc.*) with the intention of studying the variation in performance under real-world conditions. More specifically, the experiments involved measuring the call quality by varying:

- **Tor circuits:** Involved measuring quality across a large number of circuits (6650 unique relays) created using the regular Tor client program.
- **Geo-location of communication peers:** Involved measuring quality by instantiating several calls, varying the location of the calling peers.
- **Type of anonymity achieved:** Involved measuring quality while achieving one-way and two-way anonymity, in accordance with the use cases already described in Sec. 2.4.
- **Circuit lengths:** Involved measuring quality over two-hop circuits. By default, Tor circuits are built using three relays.
- **Codecs:** Involved measuring quality by varying the the call codecs used.
- **Call duration:** Involved measuring quality when call duration was varied.
- **Type of relays used:** Involved measuring quality when using bridges instead of public Tor relays.

We also measured the voice call quality for few popular voice calling apps such as `Telegram` [45] and `Skype` [25], when used over Tor. Users may choose to rely on using these already popular and familiar apps, instead of having to setup V-Tor and M-Tor. Additionally, we also conducted a user study involving 20 participants who rated calls via Tor. Most of experimental results are presented in the next section (Sec. 4). A small fraction, addressing important concerns related to VoIP performance, *e.g.*, impact of call codecs, Tor bridges, *etc.* are presented in Sec. 6.

3.3 Implementation Details

Host configurations: All machines of the in-lab experiments used Intel Core i5 8th gen CPUs with 8 GB of RAM. The hosts used in the experiments involving

public Tor relays were hosted on Digital Ocean's cloud based infrastructure, distributed across seven countries. These machines were equipped with Intel Xeon 2.2 GHz single core CPUs and had 2 GB RAM. Since the latter experiments required initiating only a single call at a time, we did not require machines with higher memory.

Tor configuration: In the basic configuration, all the caller and callee hosts had the latest Tor v0.3.9 installed. We used the Tor `stem` python library [50] to ensure that (1) only a single circuit was enabled at a time (2) the performance metrics (such as bandwidth measured using `iperf`, RTT using `ping`, *etc.*) were measured for the same circuit through which the call was performed.

Communication peers: For V-Tor experiments, the end hosts were installed with OpenVPN [27] v 2.4.7. The VoIP traffic generated was encapsulated through OpenVPN client and transported via Tor (by specifying the SOCKS port in the OpenVPN configuration). The clients used the python based SIP client, `pjsua` [31], a command-line softphone, to automate SIP calls. The module supported audio playout and recording.

VPN/SIP server: The V-Tor setup uses a machine configured to be a VPN as well as a SIP server. OpenVPN [27] v 2.4.7 was configured to work as the VPN server. It forwarded the encapsulated SIP calls to the SIP server. Freeswitch PBX [40] was used as the SIP server for handling SIP calls. We used different configuration files that come with Freeswitch to make call extensions, route calls, select codecs, *etc.*

Popular VoIP apps: Among all the popular apps, Telegram is the only one that provides user APIs. Still, we require to overcome several challenges to measure the performance of Telegram calls over Tor.

Firstly, the API provides only instant messaging automation facility. We thus mined the source code and discovered that it relies on `libtgvoip` [43] for making voice calls. We used this library for automated calls. Secondly, the library uses UDP for transporting VoIP. We thus modified it to enforce transporting voice calls via TCP streams. Thirdly, `libtgvoip` is configured, by default, to play audio clips endlessly in a loop. We applied appropriate modifications to control the playout duration, to suit our calls. Fourthly, we also required modifying the library so as to synchronize the call setup and termination events with starting and stopping of voice recording. This synchronization is required for accurately measuring PESQ. Additionally, pyrogram [44] redirected Telegram traffic to the Tor SOCKS port.

Other popular voice calling apps like Skype poses two challenges—*viz.,* absence of resources like APIs or source codes to aid call automation, and ability to redirect traffic through Tor by configurable SOCKS interface. To overcome the first obstacle we initially synced the caller and callee machines. Thereafter once the call was initiated, the caller plays out the audio clip using `mplayer` [26], while the callee captured and recorded the call audio directly from the sound card using pactl [32] utility. To overcome the second challenge, we used OpenVPN to encapsulate and redirect Skype call traffic to the Tor SOCKS interface.

4 Measurement Results

In this section, we describe the experiments conducted to test the performance of anonymous calls over Tor, and their corresponding outcomes. We begin by enlisting some common steps we followed while conducting the experiments:

- In all our experiments, a caller host played out an audio clip containing 30 s of human speech. It was encoded and transported, via a unique Tor circuit, to the callee. The callee recorded the audio, which is later used for computing PESQ.
- For every call, we recorded the network traffic through `pcap` files, and also measured various network performance attributes of the Tor circuit (through which the call is performed) like available bandwidth and RTT using `iperf` and `ping`, respectively. The `ping` utility was run during a call, as it is not a bandwidth intensive test. On the contrary, `iperf` (bandwidth intensive) tests were conducted after the completion of the call, so that it does not have any impact on the call quality.
  For the in-lab experiments also, we measured the stream bandwidth using `iperf`.
- PESQ score for every call was calculated by comparing the original (one played out at the caller) and recorded (at the callee) audio clips. Any score above three was considered good [15].
- One way delay was also calculated for the duration of the call. We used `ping` to calculate OWD. As per ITU guidelines for international calls [16], the upper limit of OWD for acceptable call quality is 400 ms.
- We ensured that for a call, all the performance metrics were measured for the same circuit through which the call was instantiated.

In every experiment, the above steps were repeated for each call. Thereafter, we analyzed the measurements and performance metrics across all these iterations.

4.1 In-Lab Experiments

We performed these experiments, with an intent of measuring call performance under different testing conditions, while fully controlling network link capabilities and background cross-traffic. To establish the baselines, we created three test scenarios for the setup. These involved: (1) Direct SIP calls (2) SIP calls over VPN tunnel (3) SIP calls through VPN over Tor (V-Tor).

All these experiments followed the setup described in Fig. 2 For the scenarios where Tor was not used, the nodes between caller and callee (in Fig. 2) merely functioned as routers. This was done to minimize any biases in performing experiments, by ensuring that the packets traverse the same number of hops[4].

Experiments using V-Tor setup: We started by initiating VoIP calls over all the three setups and computed their respective PESQ scores and OWD values. The capacity of the link between the caller and callee was 100 Mbps. The measured PESQ score averaged across 100 individual samples was the same for all the three scenarios *i.e.*, 4.5. Whereas OWD was below 50 ms. This result established that for a single call, with no competing cross-traffic, the overheads introduced by the VPN and Tor had no significant impact on the call quality. We additionally observed that the available bandwidth requirement for a single call in all the three scenarios was no more than 120 Kbps (ref. Tab. 1). As expected, direct calls transmitted at the lowest rates. Additional overheads due to the headers introduced by VPN and Tor progressively increased the bandwidth requirements.

Call category	Bandwidth (Kbps)
Direct SIP call	84
SIP call via VPN	$\approx$ 108
V-Tor	$\approx$ 120

Table 1. Baseline bandwidth (in Kbps) requirement of VoIP in different scenarios.

Next, to understand the impact of cross-traffic on VoIP call quality, we initiated VoIP calls in the presence of cross-traffic. The experiments were carried out for three link bandwidth configuration—2 Mbps, 5 Mbps and 10 Mbps. Studying the performance under cross-traffic, for different link bandwidth would help us understand if the observed behavior is consistent or not. These experiments were specifically conducted for the VPN via Tor (V-Tor) setup. Further, these lab experiments provided us insights on the number of calls that could potentially be made under varied network conditions on the real-Tor network.

Thus, we gradually introduced the cross-traffic by increasing the number of parallel file downloads (using `wget`) from another client that shared the link with the caller. We made sure that the cross-traffic was in the direction of call, to ensure adequate cross-traffic contention. We measured the degradation in the call quality, by computing the average PESQ score, for every new parallel connection introduced. The cross-traffic was gradually increased such that it utilized the total link capacity from 5%, to 10%, and then all the way up to the point where the call under consideration received less than 120 Kbps of the total available bandwidth. At this point, we observed a sharp decline in PESQ for the call (*i.e.,* 2.3). This corresponds to unacceptable call quality. Our findings are summarized in Tab. 2.

Competing Streams	Link Bandwidth	Available Bandwidth Per Stream	Call Requirement	PESQ Score
< 75	10 Mbits	> 133 Kbps	120 Kbits	> 4.2
80	10 Mbits	125 Kbits	120 Kbits	$\approx 3.4 \downarrow$
> 85	10 Mbits	< 117 Kbits	120 Kbits	$< 2.3 \downarrow\downarrow$

Table 2. Analysis of V-Tor under the presence of competing non-VoIP (web or file downloads) cross-traffic.

We then performed experiments, where the background cross-traffic constituted of other VoIP calls. Similar to the previous experiment, we performed this test for three different link capacities (2, 5 and 10 Mbps) for the V-Tor setup. The results of the 5 Mbps link bandwidth test are summarized in Tab. 3. We obtained similar behavior for the other two bandwidth categories.

Competing VoIP Calls	Link Bandwidth	Available Bandwidth Per Call	Call Requirement	PESQ Score
< 35	5 Mbits	> 145 Kbps	120 Kbits	> 4.2
40-43	5 Mbits	128 Kbits	120 Kbits	$\approx 3.3 \downarrow$
> 43	5 Mbits	< 120 Kbits	120 Kbits	$< 2.3 \downarrow\downarrow$

Table 3. Analysis of V-Tor under the presence of competing VoIP cross-traffic.

The results indicate that when the contention on the shared link increases, the PESQ drops. The PESQ metric is very sensitive to the impact of even minor network drops or delays. Even a small increase in contention, *e.g.,* only five additional download streams reduce PESQ from 4.2 to 3.4 (ref. Tab. 2). Corresponding to increased contentions, the available bandwidth for every stream (including VoIP) drops. The constant bit-rate voice traffic of 120 Kbps suffers significant distortions that may, however, have little impact on the

4 Additionally, removing these hops do not have any observable impact on our results. We kept them just to have uniformity among our experiments.

non-VoIP flows. This held when the cross-traffic was non-VoIP as well as when it was VoIP.

These in-lab measurements indicate that, a client should be able to conduct good quality calls, if the constructed circuit provides a bandwidth of above 120 Kbps. This should hold true on the real Tor network as well. *E.g.,* if a circuit has an available bandwidth of about 1.2 Mbps, then it is capable to simultaneously support a maximum of 10 VoIP clients (with acceptable call quality).

4.2 Experiments Involving Public Tor Relays

Having obtained good performance in in-lab tests, we went ahead to evaluate the performance of voice calls over public Tor network. We expected the results to vary significantly due to the dynamic nature of competing cross-traffic and network conditions over the Internet.

We began with our pilot study that involved a client host (caller), positioned in our university, establishing VoIP call to a cloud hosted peer (callee). The call traffic was transported via Tor. We sequentially started 100 calls, each transported via a different Tor circuit, and measured the call quality by computing PESQ score. To our surprise, for both the setups we observed an average PESQ of 3.8 and an acceptable OWD of 280 ms. Even after 1000 calls, each transported via a freshly created Tor circuit, we observed very similar performance measures (PESQ $\approx$ 3.86 and OWD $\approx$ 273 ms).

However, one may argue that our positive results might have been a small anomalous fraction. These may have been different from the bulk of poor outcomes that may have led others to deem Tor as unfit for VoIP. In order to test that this was not a fluke, we conducted a longitudinal experimental study covering diverse scenarios. The experiments are described below.

4.2.1 Caller anonymity: co-located voice server and callee (Scenario I).

We begin with the fundamental scenario where a caller is positioned in a censored network and wishes to call someone who is beyond the censor's control. The caller makes calls to the callee, through the public Tor network, using setups similar to one shown in Fig. 2. It is assumed that callee runs a publicly accessible VPN server (for V-Tor) on its host.

In our experiments, we chose seven individual cloud machines as callers, and three other as callees. Each of these was selected from Europe, N. America, and Asia. For every caller–callee pair we made 1000 calls using the V-Tor setup. In each case, we measured the PESQ and OWD. A total of 42000 calls were made. The average PESQ and OWD across all measurement was 3.88 and 217 ms, respectively.

By default, Tor circuits are three hops long. To reduce the potential impact of hop length on performance, we repeated the said experiments by making calls over two-hop circuits. This did not led to any significant impact on the PESQ (3.90). However, as expected, the OWD reduced to 205 ms. The CDF of PESQ scores obtained is depicted in Fig. 4. Results clearly show PESQ above 3 in over 93% of the calls.

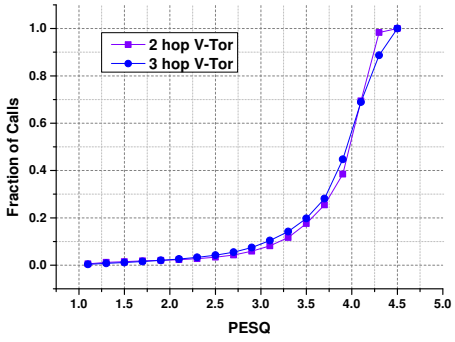

Fig. 4. V-Tor: CDF of PESQ for Caller Anonymity when server is co-located with callee (Scenario I).

4.2.2 Caller anonymity: separate VPN/voice server (Scenario II).

There are, however, certain limitations of Scenario 1. Firstly, the setup requires a publicly accessible VPN server, which may be infeasible when the callee is behind a NAT. Secondly, there may be cases where multiple whistleblowers or covert reporters (*i.e.*, several callers), communicate to callees working for a common organization. In such cases, having a commonly shared VPN/SIP server, with high availability, supporting features like voicemail, removes the need for the callee to be always online. Thirdly, it reduces the hassle for every callee to port VoIP server to different platforms.

Therefore we considered an alternative setup where the VPN / SIP server and callee were not hosted on the same host. They were distributed among seven different cloud hosts, positioned across Europe, N. America and Asia. This separation may incur higher OWD between the communication peers, due to the intervening network between the VPN/SIP server and the callee, thus impacting call quality.

To test this, the caller made 1000 individual calls (through Tor) to every callee (seven locations), via VPN/SIP servers (three locations). Similar to the previous experiment, a total of 42000 calls were conducted.

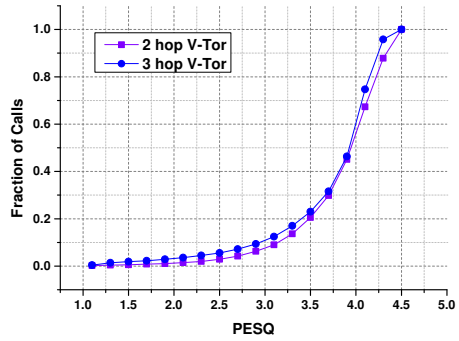

Fig. 5. V-Tor: CDF of PESQ for Caller Anonymity when server is separately hosted (Scenario II).

We observed acceptable quality with average PESQ 3.81 and average OWD 270 ms, slightly higher than the previous scenario.

Here again, we further tried to optimize the performance using shorter 2-hop circuits. We saw the average PESQ increased to 3.91, and the average OWD reduced to 210 ms. The results are presented in Fig. 5.

The CDF of OWD for V-Tor is depicted in Fig. 6. Evident from the results, we observed PESQ above 3 and OWD less than 400 ms in over 92% of the calls.

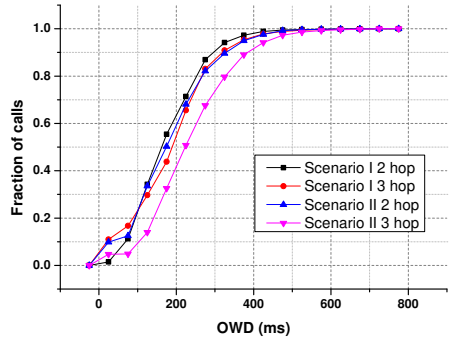

Fig. 6. V-Tor: CDF of OWD variation for Caller Anonymity in both Scenario I and II.

4.2.3 Caller and Callee (two-way) anonymity (Scenario III).

There may be cases where both the caller and callee are positioned in censored networks. In such cases, they may connect via Tor to a VPN/SIP server placed outside their respective censors' jurisdictions. Their calls would be routed via their individual Tor circuits. We thus tried to observe the impact of such scenarios (traffic traversing two circuits) on the overall call quality as the additional network hops may increase OWD.

Similar to previous experiments, we varied the caller and callee locations across seven countries, while the VPN/SIP server was distributed across three. Each caller initiated 1000 voice calls to a callee, resulting in a total of 42000 calls. For V-Tor we observed an average PESQ of about 3.2 with 81% calls above PESQ 3. However, the average OWD, as expected due to the increased network hops, was about 458 ms. This is slightly above the acceptable limit.

We thus tried to optimize performance by using shorter two-hop circuits. We hence repeated the above tests using two-hop circuits and observed a reduction of the average OWD to 396 ms. The results are shown in Fig. 7. In general, for such scenarios, regular three-hop circuits incur higher OWD, compared to two-hop ones. Hence, two-hop circuits seem a better choice for such cases. However, the results of our user study (ref Sec. 4.4) shows that users did not have any noticeable performance impact (due to the delay introduced) when both the users conversed via Tor for two-way anonymity.

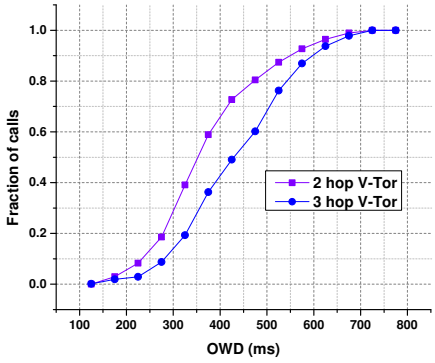

Fig. 7. CDF of delay for V-Tor setup when two-way Anonymity was achieved (Scenario III).

To summarize, in all the three scenarios over the public Tor network we observed good call quality (PESQ >3 and OWD <400 ms) in about 85% cases.

We use 400 ms as a threshold for good call quality following the ITU recommendation for international calls [15]. However, these recommendations also reported some user dissatisfaction even when OWD was between 300 ms and 400 ms. On analyzing the results, we found that more than 80% of calls had OWD below 300 ms. This has been further analyzed in detail in the Appendix. B.

4.2.4 Experiments involving popular apps

Next, we evaluate the performance of two popularly used VoIP apps, Telegram, and Skype, when running over Tor. Evaluating these apps would be beneficial from the usability point of view as most users generally use these apps for their day to day tasks. Hence using them for anonymous calls would be relatively easy, as they would not require installing V-Tor or M-Tor setups. However, the users might not be completely secure or

anonymous, when using these apps as the app maintainers would know the calling parties.

In this experiment, we instantiated 1000 consecutive calls using each of these apps (for both the setups) and computed their call quality. The average PESQ score was 3.8 and 3.54 for Telegram and Skype, respectively. Skype had more than 80% calls with PESQ score more than 3, whereas Telegram had approx. 85% above 3. Overall, there was not much difference in terms of call quality between the two applications.

In our results, we observe that popular voice calling apps perform acceptably well over Tor.

4.3 Direct Calls

In the previous subsections, we have established that users in the majority of the cases would obtain good call quality when calls are performed over Tor. However, it would be interesting to compare the performance when calls are instantiated with and without Tor to understand the relative change in call quality.

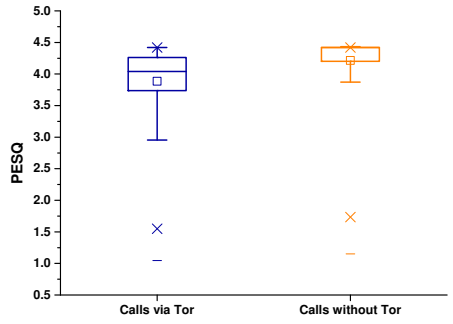

Fig. 8. Comparison of call quality in terms of PESQ obtained with and without Tor.

Hence, we performed direct calls involving both the setups. In the V-Tor setup, the calls were carried via VPN, and for M-Tor they were carried out directly using mumble. We instantiated 1000 calls for each scenario described in earlier subsections. The results are shown in Fig. 8. As can be seen from the results, there is an expected relative drop in performance when moving from non-Tor to Tor setups. This is obvious as there are expected to be much fewer distortions on non-Tor setups. However, the performance of calls via Tor is good enough for the thresholds of performance metrics used in our study (OWD <400 ms and PESQ >3).

4.4 Users' Perspective

In our research, we conducted an extensive experimental study to adjudge the quality of voice calls via Tor. The study relied on two standard metrics *i.e.*, PESQ and OWD. These metrics are robust in judging the call quality. In fact, PESQ was established as a substitute for subjective tests (as already discussed in Sec. 2.2).

However, in all our experiments, we conducted calls with a maximum duration of 30 s. This is because, PESQ does not support the evaluation of calls whose duration is longer than 30 s [34]. Moreover, the PESQ score evaluation is a non-linear function, with respect to call duration. Hence, the mean of multiple samples of 30 s will not correspond to the PESQ of the actual combined duration call, as clearly stated by ITU [34]. Also, as per ITU recommendations [34] and PESQ [35] draft, a duration of 8 s - 12 s is sufficient to judge the call quality of the channel under test. However, one might argue that the real users' experience might be different for calls above 30 s duration, as these are not tested directly with PESQ. Therefore, we conducted a user study that involved human subjects evaluating the call quality with prior approval from ethical review committee (ref. Appendix B).

The user study involved 20 participants[5], which were randomly divided into five groups of four participants each. The groups were given individual sets of calls, each containing three different recorded calls. These calls were conducted over the Tor network, using setups described in Subsection 3.1

The first call was 30 s long, and the remaining were two and four minutes long. We calculated PESQ for the 30 s call, and recorded OWD and jitter for longer duration calls (as PESQ cannot be calculated for calls longer than 30 s). The users listened to these recorded files and gave an Absolute Category Rating (ACR) [6] in the range of $1-5$. We then calculated the Mean Opinion Score (MOS) by averaging the score given by all four participants of each group (results of which are summarized in Tab. 4). The users reported an average MOS of 4.25, 4.5 and 4.0 for the 30s, 2 mins, and 4 mins calls, respectively. These results show that, in general, users reported good call quality over Tor.

Comparing MOS with network attributes: We confirmed the aforementioned MOS values (obtained from users) against recorded performance metrics. For the 30s call, we compared the MOS values with the corresponding PESQ scores. We observed an average MOS of 4.25, and a correlated average PESQ of 4.2,

5 The location of participants does not have much effect as Tor clients by default select relays in varied locations. However, our participants were spread across three countries.

6 This is in accordance with ITU guidelines [33] for rating calls in subjective tests.

Group No.	MOS 30s	2min	4min
I	4.5	4.5	4
II	4.5	4.25	4.5
III	3.5	4.75	4.25
IV	4.25	4.5	3
V	4.75	4.5	4.25

Table 4. MOS by different user groups for varied call length.

thus supporting our observations. Similarly, for the remaining two calls, we found that OWD and jitter were well within bounds for "good quality" (OWD < 400 ms and jitter < 30 ms). The average OWD was ≈ 278 ms, and jitter was about 24 ms.

All the above experiments involved the users listening to recorded calls. Thereafter, we went a step ahead, and asked ten users to converse daily via Tor for usual conversations. It must be noted that both the users connected via Tor, simulating Scenario III (two-way anonymity). Users reported a score for each of the calls. Call length varied from 1min to a max of half an hour. This experiment was conducted for about 15 days, and users in the majority of the cases reported good call quality comparable to that achieved via popular VoIP applications rating an average MOS of 4.1.

The results in this user study further strengthen the claim about obtaining adequate call quality for anonymous calls over Tor.

5 Insights from Measurements

We now present explanation of our experimental results, along with other interesting insights we observed from these results.

5.1 Overall Performance Analysis

Computing PESQ involves the comparison of the original audio clip, as played out by the caller, with what is recorded at the callee. Effects of network delays, jitters, and losses, reflected in the recorded audio, are captured by this metric. Variations in network conditions, like increase in contentious cross-traffic, leads to an increase in the drops and delays, and thus negatively impacts the perceived quality (and thus PESQ). Besides PESQ, such contention also impacts other network performance metrics like OWD, RTT and available bandwidth.

Our overwhelmingly positive results, with PESQ above 3 and OWD under 400 ms in 85% cases are indicative of relatively low network contentions that can impact VoIP call quality. VoIP calls are encoded at low sending rates (<120 Kbps) and thus require low available bandwidth. Further, in $\approx 90.6\%$ of our Tor circuits, we observed adequate available bandwidth (> 1 Mbps), as reported by `iperf`. About 95% of these 90.6% circuits supported calls with acceptable performance. This indicates that circuits with sufficient available bandwidth improves the chances of call obtaining good perceptual quality. This can be further understood by analyzing the 90.6% circuits where we measured over 1 Mbps bandwidth. We tabulate the frequency of these circuits, along with their corresponding PESQ scores. As evident from Tab. 5, for calls where we obtained good perceived quality ($PESQ > 3$), the frequency of occurrence of circuits where available bandwidth was more than 1 Mbps was also very high. Similarly, we also observed that for bad quality calls ($PESQ < 3$), the instances of circuits obtaining a bandwidth above 1 Mbps were relatively low.

PESQ	1-2	2-3	3-4	4-5
Frequency	6K	22K	162K	259K

Table 5. Variation of frequency of Tor circuits (>1 Mbps bandwidth) with PESQ of calls via them.

In general, we observe that with an increase in network contention, both call quality and available bandwidth decrease. This is evident from Fig 9. As incidences of high PESQ coincide with cases when the recorded available bandwidth is high, and vice versa.

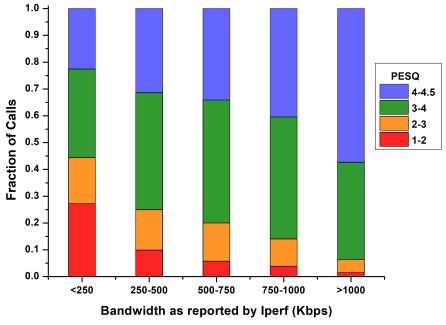

Fig. 9. Fraction of PESQ scores at different available bandwidths.

Impact on performance over time: Additionally, we also analyzed whether the performance of VoIP calls changed over time. For this, we randomly sampled 100 calls from our measurements and analyzed the distribution of calls based on quality. We repeated this experiment several times, and in *each* iteration, we observed that more than 85% calls always observed good perceived quality. This indicates that the overall distribution is uniform, and there was no observable change over time.

To ascertain the above observation, we plotted a graph incorporating results from our complete dataset. We draw a box plot consisting of variation in PESQ

scores for different months. The whiskers in the plot represent 10 and 90 percentile values. The 'x' represents the 99 and 1 percentile values, with '-' representing the min and max values respectively[7]. As evident from Fig. 10, we did not obtain any significant change in performance over time. Thus depicting that there was no observable change in performance over time.

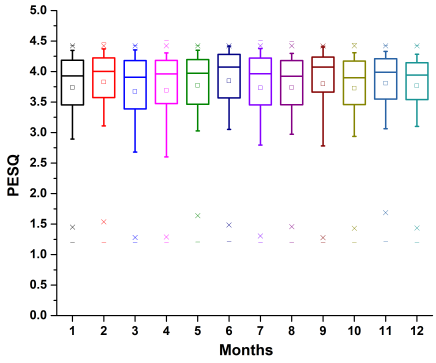

Fig. 10. Change in performance over time.

5.2 Performance Dependence on Types of Relays

We next analyzed whether different type of relays (*viz.*, Guard, Middle or Exit) along with their frequencies of occurrence in circuits, had any observable impact on the VoIP call quality over Tor. To begin with, we distributed the relay frequency into three bins — low (< 10), moderate ($150 - 200$) and high (> 500). Then, from each group, we randomly selected a few entry, middle, and exit relays (around ten each) and manually inspected the PESQ scores of the calls involving them. For each type of relays (in all frequency bins), we observed PESQ scores ranging from $1 - 4.5$, with the majority of them being over 3. Overall, there was no obvious difference in the distribution of calls with different PESQ scores. This observation indicates that PESQ neither depends on any specific type of relay, and nor on their corresponding frequency of occurrence.

To further ascertain our claims and to obtain a comprehensive picture for all our measurements, we plotted the distribution of PESQ scores corresponding to all guard, middle, and exit nodes. The box plot for exit nodes is shown in Fig. 11. As evident, the PESQ values show no dependence on the frequency of occurrence of Tor relays. Corresponding to both less (< 50) and more frequently appearing relays (> 500) we observed PESQ above 3 for a large fraction of calls. The trend was similar for guard and middle relays.

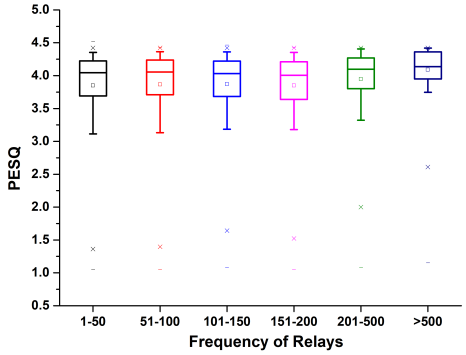

Fig. 11. PESQ of individual calls vs frequency of exit relays.

Tor churn analysis : Tor relay churn is defined as the rate of relays joining or leaving the network from one consensus to the other (according to Tor Metrics [48] and Winter *et al.* [53]). To that end, we determined whether Tor relay churn had any impact on our results. We calculated the monthly relay churn[8] for the entire duration of our study (ref Fig. 12). It is evident from Fig. 12 that relay churn was low (an avg. of $\approx 0.2\%$). Further, we also observed that more than 85% of our measured VoIP calls had acceptable quality. Our overall results thus indicate that such a low value of churn has an insignificant impact on call quality.

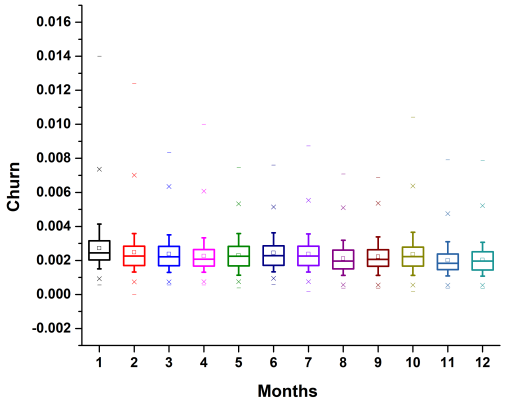

Fig. 12. Tor relay churn for the entire duration of our study *i.e.*, 12 months.

However, one may argue that a large proportion of exit relays might not support the default ports used

7 All subsequent box plots follow the same style as described.

8 The Tor consensus is updated every hour; in one month it is updated around 720 times. The individual box plot corresponds to the change in consensus of these 720 values. Thus, there are 12 box plots each corresponding to a different month.

by VoIP applications (SIP or Mumble) involved in our study. This might lead to bias in the churn analysis, as we may be considering only a small fraction of all the available exit relays that support VoIP calls. Thus, in all our experiments, we configured our VoIP servers to listen on port 80 and 443 as these are generally allowed on the majority of the exit relays.

Further, we studied the prevalence of exit nodes supporting VoIP applications by default and compared them with those who support port 80 and 443. For this, we analyzed the exit relays in the Tor consensus files for the duration of our study. First, we identified the number of exit relays allowing the default VoIP applications ports (64738 for M-Tor and 1194 for V-Tor), and also the ports used for our study (80 and 443). Once we obtained the number of exit relays allowing these ports, we added their corresponding bandwidth weights and divided it with the cumulative bandwidth weight of all the exit relays. This gave us a normalized value of the bandwidth weight of these exit relays (that allow the aforementioned ports). The results are depicted in Fig. 13. As evident from the results, almost all the exit

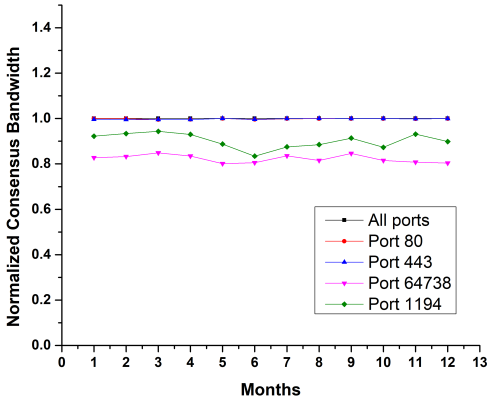

Fig. 13. Analysis of the bandwidth coverage of exit relays when using default VoIP application ports (64738 for Mumble and 1194 for VPN), compared to when port 80 and 443 were used.

relays supported port 80 (99.9 %) and 443 (99.6 %). On the other hand, 90.1 % and 83.4 % relays allowed the default VPN and Mumble ports, respectively. Thus, it indicates that our results did not rely on some specific set of exit relays. Moreover, a great fraction of exit relays (> 80%) would support our VoIP applications by default. These results, along with the churn analysis conducted previously, depict that Tor relay churn did not have any significant impact on our results.

Overall, the analysis in this subsection clearly indicates that the performance was not dependent on any specific type of relay. The apparent independence is because, for a call to be of good quality, we require less cross traffic contention, and bandwidth of about 120 Kbps for the entire call duration. The prevalence of such an ecosystem naturally on Tor has already been argued in Sec. 5.

6 Discussion

In this section, we address concerns like how VoIP performs over Tor when using bridges, using different codecs, *etc.*

Call quality over Tor bridges: Tor bridges [47], are unadvertised entry (guard) relays, whose information is closely guarded. They are conservatively distributed either through *BridgeDB* [1], or covertly via out-of-band means (*e.g.*, emails). Censors may identify and filter Tor traffic using entry node IP addresses and (or) port numbers. User residing in such networks may use bridges to access other Tor relays and set up the circuits.

Hence, to study the impact (if any) when using a bridge, we performed 1000 measurements, using the V-Tor setup. In this experiment, we used a bridge to connect to a new Tor circuit each time. The average PESQ score was about 3.7. In about 85% calls, we observed good performance (PESQ >3.0 and OWD <400 ms). It must be noted that, we restricted our measurements to a single bridge node as they are scarce.

Impact of codecs: Codecs define the way audio is encoded and decoded to transmit them as packets. Hence we measured the impact of different codecs on call quality. The calls for this experiment were conducted over the real Tor network. We used some popular codecs

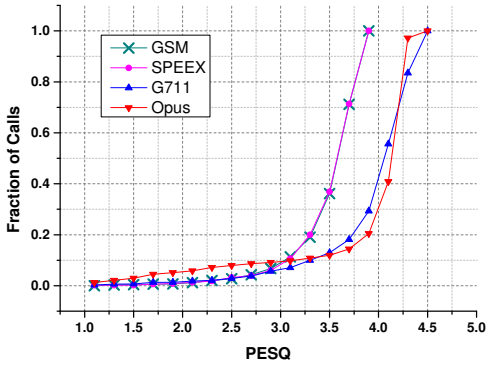

Fig. 14. CDF of PESQ scores when different codecs were used.

for our evaluation, *viz.*, *Opus, G.711, Speex, and GSM*. The codecs were selected with the help of configuration changes in the Freeswitch SIP server. We conducted 1000 calls corresponding to each of these codecs and measured their PESQ scores (ref. Fig. 14). We observed that the fraction of calls with PESQ above 3

were roughly equal (≈ 85%) in all the cases. Thus any of the tested popular codecs could have been used for initiating good quality calls. However, GSM and Speex are lossy codecs, compared to lossless codecs like G.711, and thus may not provide very high quality calls (PESQ >4) [14]. Besides, such lossy codecs encode at lower bitrates compared to the lossless ones[9]. Thus, they may be used for Tor circuits with low available bandwidth to receive adequate call quality.

Thresholds for PESQ: The MOS scale proposed by ITU delineates perceived performance through sharp integral differences. *E.g.*, two corresponds to "annoying but usable" and three corresponds to "fair". However, it does not extend such annotations for values in-between. Further, upon listening to a few calls manually where PESQ was between 2.8 and 3.2, we observed no audible differences. Moreover, recent studies [21] indicate that humans may be unable to differentiate the quality of samples, when the difference between their MOS scores is less than 0.4.

In our study, we *conservatively* selected a PESQ over 3 as "acceptable", and anything below it as not (in accordance with ITU). We also observed a significant 7% cases which maybe categorized as "annoying but usable." Finally, we also observed about 8% with PESQ under 2. ITU classifies these as "totally unacceptable." Thus, we believe that actual user experience may be even better than what our study reports (as evident from the conducted user study in Subsec. 4.4).

Coverage of Tor relays: We recorded and analyzed Tor circuit information for all our experiments. We now present some interesting insights we observed from this analysis. A total of about 600 K Tor circuits were created during our study.[10] These circuits involved a total of 6650 unique Tor relays. Prior research (Rizal *et al.* [36]) reportedly used only about 298 relays that too restricted to Europe.

Selection of caller-callee endpoints: Throughout our experiments, we used endpoints either as cloud hosts or in-lab machines. All of these were sufficiently provisioned for a voice call with more than adequate bandwidth at the caller-callee end. However, one might argue that our results might be biased as all our endpoints may have sufficient bandwidth for VoIP calls. But in general, the performance bottleneck was introduced at the Tor relays. As already described in Sec. 5, there were a significant number of Tor circuits which observed a low bandwidth (< 1 Mbps). Hence, even if the endpoints are well provisioned, it does not bias our results, as mostly the Tor relays were the bottlenecks.

9 GSM and Speex encode at bitrates <40 Kbps whereas G.711 encodes at 84 Kbps.

10 The number of Tor circuits are slightly higher than the total number of experiments as the two way anonymity experiments involve creating two Tor circuits for a single call.

7 Conclusion

Real-time anonymous VoIP calls are of interest to privacy and anonymity conscious citizens, whistle-blowers and covert reporters, *etc.* However, existing research on performance evaluation of VoIP calls over Tor is not comprehensive. They contraindicate transporting VoIP packets over Tor and thus favored novel architectures to support anonymous calling. Moreover, there does not exist a functional system to achieve the same. Additionally, the costs involved in recruiting volunteer operated relays (like Tor) along with users, and managing such a system, might outweigh the benefits.

Thus, it was essential to identify the causes of poor voice call quality over Tor by observing how the interplay of various network attributes (RTT, available bandwidth, *etc.*) impacts VoIP quality. To that end, we conducted a longitudinal study (spread across 12 months). It involved extensive testing of about half a million voice calls over Tor, including a user study, using various Tor circuits, peer locations, popular apps, *etc.* To our surprise, in over 85% cases, we observed good performance (PESQ>3 and OWD <400 ms), with only under 8% cases which were totally unacceptable (PESQ<2). The results of the user study also corroborate our findings. Our study is the first to demonstrate that anonymous VoIP calls are indeed possible using Tor.

8 Acknowledgements

We would like to thank our shepherd Rob Jansen and other anonymous reviewers for their valuable inputs. A special mention to Devashish Gosain, who provided essential comments and feedback throughout the project. He also helped in conducting some experiments.

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

References

[1] Tor bridges - bridgedb. https://bridges.torproject.org/.

[2] Tor metrics. https://metrics.torproject.org/.

[3] *Users guide for PRISM Skype collection*, August 2012. https://www.spiegel.de/media/media-35530.pdf.

[4] *NSA uses powerful toolbox in effort to spy on global networks*, December 2013. https://www.spiegel.de/international/world/\the-nsa-uses-powerful-toolbox-in-effort-to-spy-on-global-networks-\a-940969.html.

[5] Cangialosi, F., Levin, D., and Spring, N. Ting: Measuring and exploiting latencies between all tor nodes. In *Proceedings of the 2015 Internet Measurement Conference* (2015), ACM, pp. 289–302.

[6] Chaum, D. L. Untraceable electronic mail, return addresses, and digital pseudonyms. *Communications of the ACM 24*, 2 (1981), 84–90.

[7] Danezis, G., Diaz, C., Troncoso, C., and Laurie, B. Drac : An architecture for anonymous low-volume communications. In *International Symposium on Privacy Enhancing Technologies Symposium* (2010), Springer, pp. 202–219.

[8] Dingledine, R., Mathewson, N., and Syverson, P. Tor: The second-generation onion router. Tech. rep., Naval Research Lab Washington DC, 2004.

[9] Dingledine, R., and Murdoch, S. J. Performance improvements on tor or, why tor is slow and what we're going to do about it. *Online: http://www. torproject. org/press/presskit/2009-03-11-performance. pdf* (2009).

[10] Gerhard Rieger. *socat*, April 2009. http://www.dest-unreach.org/socat/.

[11] The Guardian. *NSA collecting phone records of millions of Verizon customers daily*, june 2013. https://www.theguardian.com/world/2013/jun/06/nsa-phone-records-verizon-court-order.

[12] Handley, M., Jacobson, V., and Perkins, C. Sdp: session description protocol. Tech. rep., 2006.

[13] Heuser, S., Reaves, B., Pendyala, P. K., Carter, H., Dmitrienko, A., Enck, W., Kiyavash, N., Sadeghi, A.-R., and Traynor, P. Phonion: Practical protection of metadata in telephony networks. *Proceedings on Privacy Enhancing Technologies 2017*, 1 (2017), 170–187.

[14] Ilias, I. S. H. C., and Ibrahim, M. S. Performance analysis of audio video codecs over wi-fi/wimax network. In *Proceedings of the 8th International Conference on Ubiquitous Information Management and Communication* (2014), pp. 1–5.

[15] ITU-T, I. Recommendation g. 114. *One-Way Transmission Time, Standard G 114* (2003).

[16] ITU-T, R., and Recommend, I. G. 114. *One-way transmission time 18* (2000).

[17] Jansen, R. *Onionperf : A utility to track Tor and onion service performance.* The Tor Project, May 2015. https://onionperf.torproject.org/onionperf.html.

[18] Jansen, R., and Hopper, N. Shadow: Running tor in a box for accurate and efficient experimentation. *Proceedings of Network and Distributed Systems Security (NDSS) 2012*.

[19] Jansen, R., Vaidya, T., and Sherr, M. Point break: a study of bandwidth denial-of-service attacks against tor. In *28th USENIX Security Symposium (USENIX Security 19)* (2019), pp. 1823–1840.

[20] Johnson, A., Jansen, R., Hopper, N., Segal, A., and Syverson, P. Peerflow: Secure load balancing in tor. *Proceedings on Privacy Enhancing Technologies 2017*, 2 (2017), 74–94.

[21] Katsigiannis, S., Scovell, J., Ramzan, N., Janowski, L., Corriveau, P., Saad, M. A., and Van Wallendael, G. Interpreting mos scores, when can users see a difference? understanding user experience differences for photo quality. *Quality and User Experience 3*, 1 (2018), 6.

[22] Le Blond, S., Choffnes, D., Caldwell, W., Druschel, P., and Merritt, N. Herd: A scalable, traffic analysis resistant anonymity network for voip systems. In *ACM SIGCOMM Computer Communication Review* (2015), vol. 45, ACM, pp. 639–652.

[23] Lightspeed gaming LLC. *Mumble*, March 2009. https://www.mumble.com/.

[24] Mani, A., Wilson-Brown, T., Jansen, R., Johnson, A., and Sherr, M. Understanding tor usage with privacy-preserving measurement. In *Proceedings of the Internet Measurement Conference 2018* (2018), pp. 175–187.

[25] Microsoft. *Skype*, August 2003. https://skype.com/.

[26] The Mplayer Project. *Mplayer*, January 2000. http://www.mplayerhq.hu/design7/news.html.

[27] OpenVPN INC. *OpenVPN*, November 2006. https://www.openvpn.net/.

[28] Panchenko, A., Pimenidis, L., and Renner, J. Performance analysis of anonymous communication channels provided by tor. In *2008 Third International Conference on Availability, Reliability and Security* (2008), IEEE, pp. 221–228.

[29] Perry, M. Torflow: Tor network analysis. *Proc. 2nd HotPETs* (2009), 1–14.

[30] Pfitzmann, A., Pfitzmann, B., and Waidner, M. Isdn-mixes: Untraceable communication with very small bandwidth overhead. In *Kommunikation in verteilten Systemen* (1991), Springer, pp. 451–463.

[31] PJSIP. *pjsua*. https://www.pjsip.org/pjsua.htm.

[32] Pulseaudio. *pactl*, June 2011. https://linux.die.net/man/1/pactl.

[33] Rec, I. P. 830: Subjective performance assessment of digital telephone-band and wideband digital codecs. *International Telecommunication Union, Geneva (Switzerland)* (1996).

[34] Rec, I. P. 862.3: Application guide for objective quality measurement based on recommendations p. 862, p. 862.1 and p. 862.2. *International Telecommunication Union, Geneva* (2005).

[35] Rix, A. W., Beerends, J. G., Hollier, M. P., and Hekstra, A. P. Perceptual evaluation of speech quality (pesq)-a new method for speech quality assessment of telephone networks and codecs. In *Acoustics, Speech, and Signal Processing, 2001. Proceedings.(ICASSP'01). 2001 IEEE International Conference on* (2001), vol. 2, IEEE, pp. 749–752.

[36] Rizal, M. *A Study of VoIP performance in anonymous network-The onion routing (Tor)*. PhD thesis.

[37] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and Schooler, E. Sip: session initiation protocol. Tech. rep., 2002.

[38] Schatz, D., Rossberg, M., and Schaefer, G. Reducing call blocking rates for anonymous voice over ip communications. In *Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT), 2017 9th International Congress on* (2017), IEEE, pp. 382–390.

[39] Schulzrinne, H., Casner, S., Frederick, R., and Jacobson, V. Rtp: A transport protocol for real-time applications. Tech. rep., 2003.

[40] SignalWire. *Freeswitch*, January 2006. https://freeswitch.com/.

[41] Snader, R., and Borisov, N. Eigenspeed: secure peer-to-peer bandwidth evaluation. In *Proceedings of the 8th international conference on Peer-to-peer systems* (2009), USENIX Association.

[42] Snader, R., and Borisov, N. Improving security and performance in the tor network through tunable path selection. *IEEE Transactions on Dependable and Secure Computing 8*, 5 (2010), 728–741.

[43] Telegram. *libtgvoip*, February 2017. https://github.com/grishka/libtgvoip.

[44] Telegram. *pyrogram*, January 2018. https://github.com/pyrogram/pyrogram.

[45] Telegram Messenger LLP. *Telegram*, August 2013. https://telegram.com/.

[46] Tor dev team. *Torocks*, December 2000. https://linux.die.net/man/8/torsocks.

[47] The Tor Project. *Tor Bridges*. https://2019.www.torproject.org/docs/bridges.html.en.

[48] The Tor Project. *The Tor Metrics Project*, January 2009. https://metrics.torproject.org/.

[49] The Tor Project. *Chutney.*, February 2011. https://github.com/torproject/chutney.

[50] Tor Project. *Stem*, March 2013. https://stem.torproject.org/.

[51] Tor Project. *Torfone*, April 2013. http://torfone.org.

[52] The Tor Project. *Simple Bandwidth Scanner.*, March 2018. https://github.com/torproject/sbws.

[53] Winter, P., Ensafi, R., Loesing, K., and Feamster, N. Identifying and characterizing sybils in the tor network. In *25th USENIX Security Symposium (USENIX Security 16)* (2016), pp. 1169–1185.

A M-Tor Results

In this section, we describe the implementation details along with the experiments performed using the M-Tor setup. The client machine configuration along with Tor's setup was similar to the V-Tor setup (described in detail in Subsec. 3.3). The detail of the other new entities introduced in the M-Tor setup is described below:

Communication Peers: For M-Tor experiments, the end-point hosts were installed with Mumble client program, configured to work in TCP mode. This enables directly sending voice traffic over TCP streams. Further, like OpenVPN, the Mumble client also allows transporting the voice traffic through Tor, by specifying the SOCKS port in its configuration. `Torsocks` [46] utility was used to transport `iperf`'s traffic over Tor. Since `torsocks` does not allow programs with root privileges, we relied on `socat` [10] tunnels for transporting `pings`.

Mumble Server: The M-Tor setup had the Mumble server (Murmur) v1.2.19 (analogous to VPN/SIP server in V-Tor setup) installed for handling voice calls. A call channel was opened for every new call. The caller and callee utilities were configured to join this call channel so that whenever a caller initiated a call, it would reach the channel and the callee could record it for quality evaluation.

Now, we describe the experiments involving the controlled setups as well the ones performed over the public Tor network. *All these experiments followed the setups similar to those of V-Tor.*

A.1 Controlled Experiments

Similar to the V-Tor experiments, we performed two different sets of tests using M-Tor in the lab environment. These involved: (1) Direct Mumble calls without Tor (2) Mumble calls through Tor. We recorded an average PESQ (measured across 100 individual calls) of 4.5 for both direct calls and calls through Tor. Here also we observed a slightly higher bandwidth requirement ($70-80$ Kbps) for calls that were transported via Tor, compared to those that were not ($50-60$ Kbps). In general, the bandwidth requirement for M-Tor was much lower in comparison to V-Tor as the underlying codec used by Mumble encodes at a lower rate.

Further, similar experiments were performed where we increased the number of parallel connections gradually, to see its impact on call quality. Here also we observed a trend identical to V-Tor experiments.

A.2 Experiments over Public Tor

We considered the same three scenarios to perform experiments using M-Tor *i.e.,* (1) Caller anonymity: Co-located voice server and callee *viz,* Scenario I (ref. Subsection. 4.2.1) (2) Caller anonymity: separate VPN/voice server *viz,* Scenario II (ref. Subsection. 4.2.2) (3) Caller and Callee (two-way) anonymity *viz,* Scenario III (ref. Subsection. 4.2.3). The CDF of PESQ scores obtained in Scenario I and Scenario II are depicted in Fig. 15, and Fig. 16 respectively. It is evident from the figure that majority of the calls obtained PESQ values greater than 3. Similarly, the average PESQ score obtained in Scenario III, was 3.8 with 85% calls above a PESQ of 3.

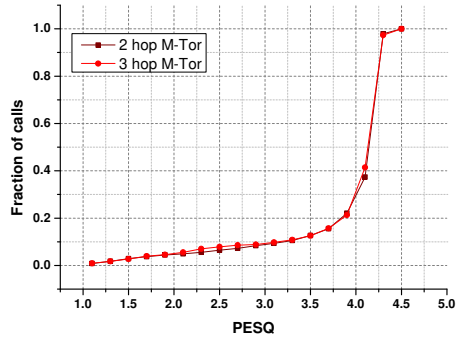

Fig. 15. M-Tor: CDF of PESQ for Caller Anonymity when server is co-located with callee (Scenario I).

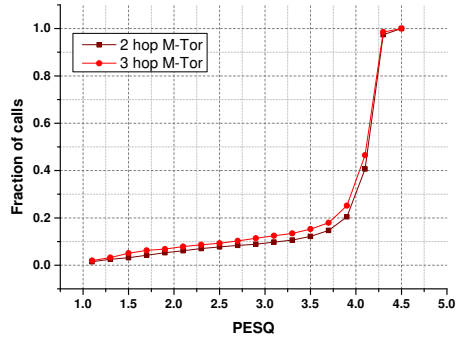

Fig. 16. M-Tor: CDF of PESQ for Caller Anonymity when server is separately hosted (Scenario II).

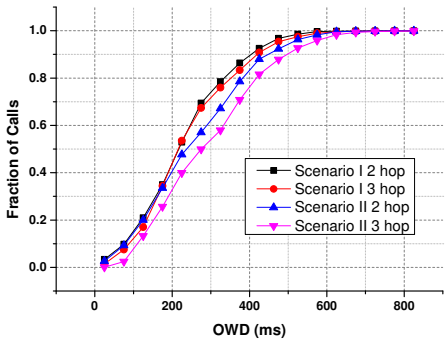

Fig. 17. M-Tor: CDF of OWD variation for Caller anonymity in both Scenario I and II.

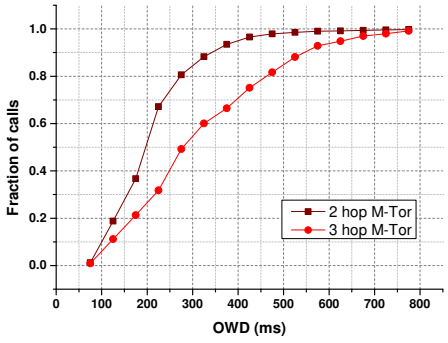

Fig. 18. CDF of delay for M-Tor setup when two-way anonymity was achieved (Scenario III).

The OWD results for both Scenario I and II (for three hop as well as two hop circuits) are shown in Fig. 17, with the results of Scenario III in Fig. 18. As evident from the results, moving to two-hop circuits in Scenario III (two-way anonymity) helped us improve the OWD significantly with more than 90% calls below 400ms in comparison to 70%.

Overall, similar to V-Tor, M-Tor proved to be capable of performing good quality calls for the majority of the cases.

B Miscellaneous Issues

VoIP applications with high bandwidth requirement: Our measurements involved testing VoIP applications that encoded at low bit-rates ($< 120Kbps$). However, these applications can be configured to encode at higher rates ($\approx$ 800 Kbps). We evaluated the performance at these higher encoding rates (200 Kbps, 400 Kbps and 800 Kbps). We ran 1000 calls when the clients were configured to encode at these rates.

As expected, an increase in the rates progressively deteriorated the performance, and thus the measured PESQ. Average PESQ scores were 3.6, 3.2 and 3.0 for 200, 400, and 800 Kbps rates, respectively. However, even at 800 Kbps, we measured PESQ of above 3 for 65% of the cases. Thus even at higher encoding rates, one can expect reasonable call quality.

Coverage of Tor relays: We recorded and analyzed Tor circuit information for all our experiments. We now present some interesting insights we observed from this analysis. A total of about 600,000 Tor circuits were created during our study.[11] These circuits involved a total of 6650 unique Tor relays. Prior research (Rizal *et al.* [36]) reportedly used only about 298 relays that too restricted to Europe.

Threshold of OWD: We analyzed the distribution of OWD values for our study to obtain deeper insights about the overall performance. Fig. 19 shows the percentage of calls which have OWD less than 50 ms, 100 ms so on till 400 ms. We notice that only about 10% of calls had a one-way delay above 300 ms and below 400

11 The number of Tor circuits are slightly higher than the total number of experiments as the two way anonymity experiments involve creating two Tor circuits for a single call.

ms, indicating only a small fraction of calls in that category. On the contrary, for about 81% of calls, the OWD was below 300 ms. This suggests that in the majority of the cases, the user would obtain satisfactory call quality with OWD less than 300 ms. Moreover, in about 28% of calls OWD was below 150 ms, which is regarded as an ideal quality call according to ITU. Thus, though we considered the ITU recommendations of OWD less than 400 ms to judge call quality, for the majority of our calls, we observed OWD below 300 ms.

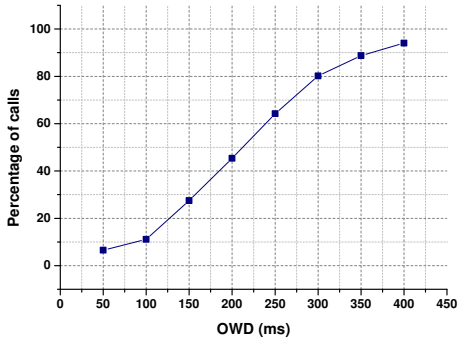

Fig. 19. Percentage of calls recorded at different OWD values.

Ethical Considerations: As described in the paper, we generated all our network traffic (*i.e.*, the voice calls) using machines in our control. We did not capture or use any third party's data/network traffic. Moreover, as our measurements were spread across a span of 12 months, and involved generating low bit-rate voice traffic ($\approx$ 120 Kbps), along with a short duration (<10 s) single `iperf` probes, we expect it to have had negligible to no impact on any un-involved Internet users' network performance.

Our user study involved human subjects in different geographic locations who heard audio samples and spoke to one another, via our setups. To the best of our knowledge the audio sample bore no information that may cause emotional or psychological trauma to the subjects involved. The quality and the contents of the clips were duly attested by the institutional research review committee that involved subjects who were not party to the research in any capacity. Further, we did not record and (or) decode, either manually or electronically, the speech between subjects, thereby preserving their communication privacy.