Experiences from Netalyzr with Engaging Users in End-System Measurement

Christian Kreibich[†]   Nicholas Weaver[†]   Gregor Maier[†]   Boris Nechaev[*]   Vern Paxson[†‡]

[†]International Computer Science Institute
{christian,nweaver,gregor}@icir.org

[*]HIIT & Aalto University
boris.nechaev@hiit.fi

[‡]UC Berkeley
vern@cs.berkeley.edu

ABSTRACT

Netalyzr is a widely used network diagnostic and debugging tool that has collected 259,000 measurement sessions to date. To use *Netalyzr*, users visit its website, download an applet that proceeds to conduct a suite of tests and measurements, and obtain a summary report detailing the findings. Along with the measurement data, for each session, we record the HTTP referrer that brought the user to the *Netalyzr* page, the level of trust the user bestowed upon the applet, and any feedback that the user voluntarily left via a form that we include at the bottom of the report page. These data sources illuminate how *Netalyzr*'s users employ the tool, and can provide insights as to how other measurement tools or user surveys involving end-host measurement could effectively involve users. We find that even with little prompting, users leave explicit comments 3% of the time and answer one or more survey questions in 17% of the sessions, reaching up to 44% of sessions during bursts of activity. We also find that significant usage of the tool comes from four types of need: (*i*) to aid in troubleshooting performance for an on-line game, often via measurement sessions conducted when requested by more sophisticated users in a help forum; (*ii*) curiosity, often exacerbated by blog postings and other mentions on high-profile websites; (*iii*) repeat visitors who arrive via a search engine that they used to locate *Netalyzr*'s website; and (*iv*) IPv6 deployment tests conducted or organized by specialists.

Categories and Subject Descriptors

C.4 [**Performance of Systems**]: MEASUREMENT TECHNIQUES

General Terms

Human Factors, Measurement, Performance, Reliability

1. INTRODUCTION

> *"My network is a nightmare."*
>
> —*Netalyzr* user, March 2010

The *ICSI Netalyzr* is a widely used network diagnosis and debugging tool, available at http://netalyzr.icsi.berkeley.edu. This publicly available service enables the user to obtain a detailed analysis of the operational envelope of their Internet connectivity, serving both as a source of information for the curious as well as an extensive troubleshooting diagnostic should users find anything amiss with their network experience. *Netalyzr* tests a wide array of properties of users' Internet access, from the network layer to applications. Its tests include IP address use and translation, IPv6 support, DNS resolver fidelity and security, TCP/UDP service reachability, proxying and firewalling, antivirus intervention, content-based download restrictions, content manipulation, HTTP caching prevalence and correctness, network and protocol-level latencies, and access-link buffering. We make the measurement results *visible* to users in the form of a test result summary webpage; indeed, one of our key motivations for developing *Netalyzr* was to provide users with an only (we hope) moderately technical summary of the measurement results. For a detailed description of the system and analysis of the measurements collected up through September 2010, see [1].

In general, a key question when conducting end-host measurement regards to what degree to involve the user in the measurement process. The design space here is wide; ranging from purely automated data collection, gathered without the user's involvement and perhaps longitudinally, to fully user-initiated troubleshooting tickets. When designing *Netalyzr*, we gave thought to this problem and chose to rely on users to decide themselves when to initiate measurement sessions; leave them mostly out of the measurement process itself; but allow them to provide additional feedback once the measurements have completed and users have had the opportunity to inspect the findings. For this latter, the test result page ends with a small feedback form via which users can provide additional information if they wish. In addition, *Netalyzr* records HTTP Referer headers to allow us to track how users arrived at the *Netalyzr* website. This information can provide valuable insights into the user's motivation. For example, users arriving via a highly technical site (such as http://comcast6.net, which recommends *Netalyzr* for IPv6 testing) likely have different motivations than users arriving via the web forum of an online game.

In this paper we present an analysis of these different sources of information regarding why users' motivations and experiences with performing the measurements. We study 8,500 of the user-provided comments in order to understand what kinds of information users voluntarily share (or feel a need to share even though the measurements already report it), and what questions, suggestions, or disagreements they find worth mentioning. Our analysis is opportunistic: we designed the user feedback form as a way for users to provide additional connectivity information and general
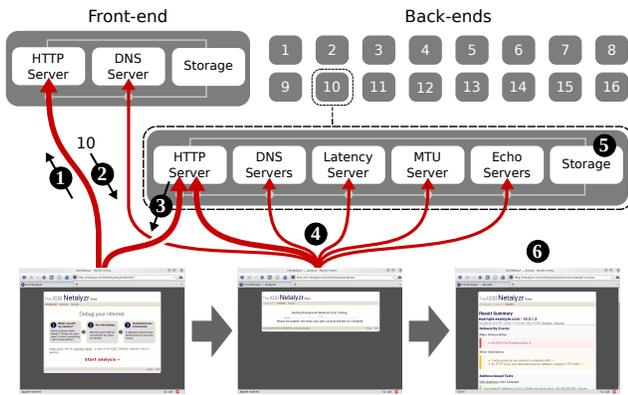
Figure 1: *Netalyzr*'s conceptual architecture. ❶ The user visits the *Netalyzr* website. ❷ When starting the test, the front-end redirects the session to a randomly selected back-end node (#10 in this case). ❸ The browser downloads and executes the applet. ❹ The applet conducts test connections to various *Netalyzr* servers on the back-end, as well as DNS requests that are eventually received by the main *Netalyzr* DNS server on the front-end. ❺ We store the test results and raw network traffic for later analysis. ❻ *Netalyzr* presents a summary of the test results to the user.

feedback on *Netalyzr*, and not as a first-order instrument to gather data for a scientific user study. Accordingly, we focus the paper on insights gained from the information users have provided us, and not on the study of alternative designs for soliciting user feedback and any diverging information this might have produced.

Apart from frequent praise for *Netalyzr*—which highlights the efficacy of engaging users by providing them with measurements they find valuable and comprehensible—roughly half of our users report on technical details such as their ISP, connection type and quality, or some particulars regarding hardware and software they employ. (A small minority of users, 6%, also leave an email address.) Furthermore, the users' classification of their networks in terms of purpose and connection technology appears accurate, as the results generally match our expectations and understanding of different network technologies.

We begin with a review of *Netalyzr*'s architecture in Section 2 and the overall user feedback mechanism in Section 3. We characterize the free-form comments in Section 4 and the properties of the reported networks in Section 5. We analyze how users access *Netalyzr* in Section 6 and discuss a smidgen of future plans in Section 7. We lightly touch on related work in Section 8 and conclude in Section 9.

2. DESIGN & IMPLEMENTATION

"If i knew what all this meant, i'd be dangerous"
—*Netalyzr* user, February 2010

When designing *Netalyzr* we sought to strike a balance between a tool with sufficient flexibility to conduct a wide range of measurements and tests, and a interface simple enough for non-technical users to find usable. To this end, we decided to base our approach on a Java applet ($\approx$ 5,000 lines of code) to drive the bulk of the test communication with custom servers ($\approx$ 12,000 lines of code), since (*i*) Java applets run automatically within most major web browsers, (*ii*) applets can engage in raw TCP and UDP flows to arbitrary ports

Figure 2: *Netalyzr*'s feedback form.

(though not with altered IP headers), (*iii*) if the user approves trusting the applet, it can contact hosts outside the same-origin policy, (*iv*) Java applets come with intrinsic security guarantees for users (e.g., no host-level file system access allowed by default runtime policies), (*v*) Java's fine-grained permissions model allows us to adapt gracefully if a user declines to fully trust our applet, and (*vi*) no alternative technology matches this level of functionality, security, and convenience. Figure 1 shows the *Netalyzr* architecture. For more details, we refer the reader to our prior work [1].

The applet can run either in *trusted* or *untrusted* mode. Depending on the browser, the Java applet runtime presents a message to the user at applet startup that asks whether the user wishes to trust the applet, showing the applet's signature as an indication as to the origin of the code. If the user confirms trust, the runtime allows the applet to conduct a more extensive set of tests, details of which depend on the runtime's configuration. The applet identifies the extent to which the user has allowed it to engage in network I/O by catching Java's security exceptions. 97% of *Netalyzr* sessions allowed the applet to have unrestricted network access.

We added referrer tracking in March 2010. Previously we could only approximate the referrer information by analyzing the HTTP logs from the front-web server. Thus, only 61% of the sessions include referrer tracking. Even with tracking, only 59% of the sessions include a referrer; presumably the remainder reflect invocation by typing the *Netalyzr* website name directly into a browser's address bar. In general, this tracking omits the initial flash crowds, but provides a good picture of sustained usage.

3. VOLUNTARY USER FEEDBACK

"Internet is one hell of a drug."
—*Netalyzr* user, June 2010

Figure 2 shows *Netalyzr*'s feedback form as presented to the user in the test results report. We have kept the form's content and layout unchanged throughout *Netalyzr*'s lifespan. It includes two radio buttons, one for the LAN-level network connection type and the other for the setting. We also provide free-form text entry fields for additional comments and the user's email address.

The feedback form comes at the very end of the potentially rather lengthy results page; the more problems the tests have identified,

#	Tag	#	Tag (cont.)
108	bugreport	227	reason
102	confusion	126	reason-gaming
380	conn-frustrated	3	reason-3rdparty-debug
14	conn-happy	10	reason-p2p
27	critique	51	researcher
125	entertaining	373	suggestion
356	foreign-language	643	technical
538	help	44	technical-agreement
170	help-buffering	568	technical-dns
5	interesting	788	technical-hw
76	investigate	1737	technical-isp
1740	location	1553	technical-link
40	location-vehicular	184	technical-objection
190	other	16	technical-othertool
2549	praise	687	technical-software
113	question	116	unclear

Table 1: The set of content tags we produced during our feedback labeling pass.

the longer the results summary. While we provide mechanisms to expand and collapse the test result descriptions, even a problem-free session's summary comprises 2–3 screen's worth of content before the feedback form becomes visible. Problematic findings are automatically expanded to alert the user, further lengthening the distance between the start of the page and the feedback box. Users may augment their feedback for up to an hour after the completion of the tests. Subsequent revisits of the summary report only shows the provided feedback, if any, in static form.

Despite this arguably tucked-away location of the feedback form, 17% of our users (roughly 45,000) left some sort of feedback. Connection type and purpose of the location are both virtually always given if the user provides any feedback. 6% left some sort of email address and 3% a free-form comment. The predominant email domains come from the major email providers (GMail, Hotmail, Yahoo).

Users were generally split nearly equally between wired (57%) and wireless networks. As we develop below, this self-reporting appears accurate. Most users who declare their intentions run Netalyzr at home (80%). Of the remaining fraction, 16% report at work, 2% declare a public setting and 2% specify "other".

4. NATURE OF COMMENTS

"This is a whole new level to 'just bang it with a wrench'."

—*Netalyzr* user, June 2009

In order to study the users' feedback, we built a web interface to present randomly selected feedback-enabled sessions for manual characterization. Starting from a broad, general set of tags we labeled all 8,537 commented sessions iteratively by hand. We labeled each comment with one or more tags, refining the broader tags (in moderation) into specializations as needed. Table 1 shows the resulting set of tags. Except for "foreign-language", the tags with hyphenation indicate specializations we added over time.

A large number of comments (3,610) include technical information. This information included items such as ISP relationship and network link properties, software and hardware versions, and local software or activity that affected results. Location and network

type tags, along with the 40 sessions within moving vehicles, enable us to evaluate the properties of different network technologies (see below).

Another common theme (2,549 users) was some form of praise or encouragement, highlighting the general potential of tapping end-user enthusiasm for conducting measurement studies. Praise without additional information, however, was rare. 373 offered suggestions for improvements or more tests. 298 had a more critical tone, either mentioning general deficiencies, offering technical objections to the results, or flagging bugs. None of these comments struck us as hostile, however. Another 102 users expressed confusion regarding the results, and 690 requested help interpreting the results.

Finally, 125 struck us as particularly entertaining, of which we selected a handful to introduce the sections of this paper.

5. NETWORK PROPERTIES

"802.11g wireless (currently at 36mbit) at far end of range through an extremely shitty thomson (spit! ack! ptui!) 787v (barf) integrated modem/router/ap/voip thing lobotomized through isp-custom firmware."

—*Netalyzr* user, November 2009

To check whether the 17% of sessions in which users had indicated their LAN connectivity and location match our assumptions about their networks, we cross-checked the feedback with observed network behavior. Our main goal with this assessment is to gauge to what degree future measurement studies might rely upon users providing accurate information.

The findings reconcile: for *Netalyzr* runs reported as running at work, we find a mean latency (as measured to our East-coast EC2 node) of 110ms; those from home showed 130ms; and those from public locations, 180ms. We see similar differences for bandwidth: 11 Mbps/8.3 Mbps download/upload speeds at work; 7.0 Mbps/1.8 Mbps at home; and 4.1 Mbps/2.7 Mbps in public settings. These figures make sense, as we would expect that business connections will tend towards both higher quality and greater symmetry.

Similarly, we find that connections reported as wireless suffer more packet drops: 54% of wireless connections experienced drops, compared with 33% of wired sessions. 11% of the wireless sessions experienced at least one transient outage (loss of ≥ 3 packets in a row during 10 Hz background probing), but only 5% for wired sessions. We also observe a distinction in the uplink buffering tests, where wired networks show a more consistent buffer distribution, and fewer extremely bad cases (Figure 3).

While characterizing the feedback comments, we also noted information about the technology used, such as satellite or cellular connections, or connectivity from moving vehicles. Figure 4 shows the differences in various technologies for bandwidth latency and loss.

Moving-vehicle technologies suffered from particularly poor performance, with very low bandwidth in both directions. Except for airplanes, they also frequently exhibited high latency and packet loss. Satellite networks also manifested problematic performance, with highly asymmetric bandwidth and (unsurprisingly) large latencies, though packet loss rates were not highly elevated.

We found better-than-expected performance for cellphone-based connections. They showed download bandwidth comparable to typical hotel networks, and did not suffer undue latency or loss considering their wireless nature. Several free-form comments offered derogatory comments concerning particular hotel networks.
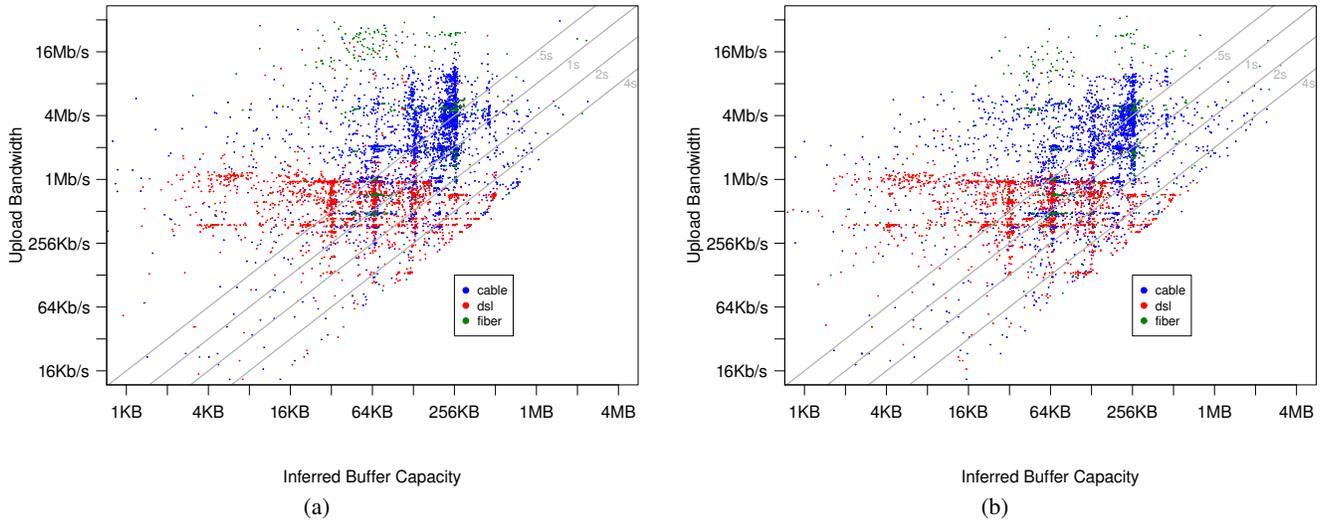
Figure 3: Uplink buffering as observed for wired (a) and wireless (b) hosts for major ISPs where we can classify the link technology used in the connection.
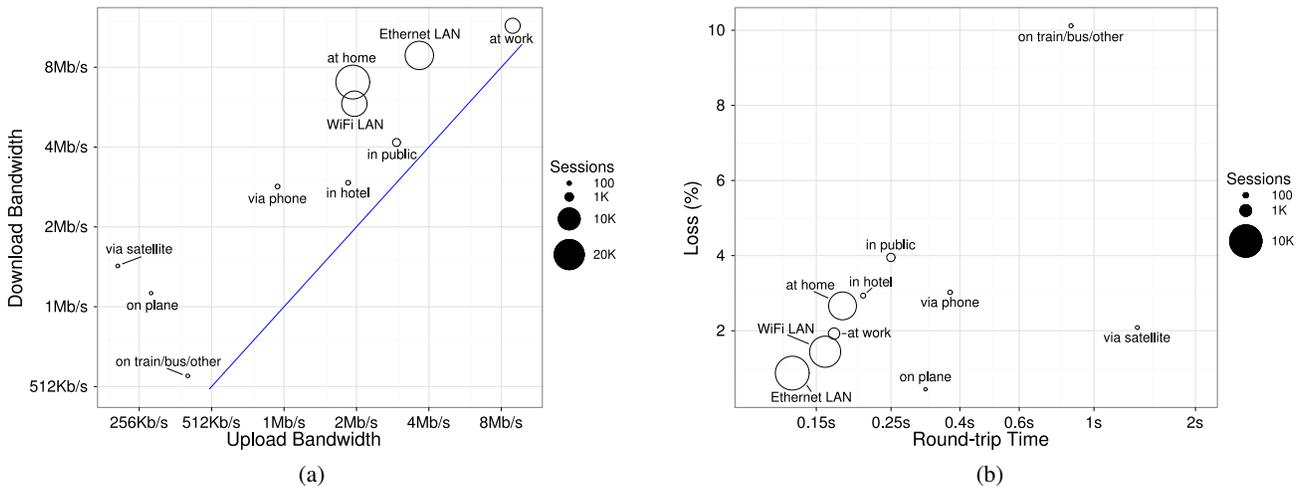

Figure 4: Uplink and downlink bandwidth (a) and latency and loss (b) for reported locations and technology types.

6. REFERRER TRACKING

"Found link to your service on El Reg (The Register— gotta luv the brits)."

—*Netalyzr* user, January 2010

Studying the HTTP `Referer` headers with which users arrive at the *Netalyzr* site allows us to track visitor origin, and also lets us draw conclusions about the users' likely motivation for running *Netalyzr*'s tests.

Our largest source of referrals to date is heise.de, with 34% of referral-carrying visits. They reported on *Netalyzr* shortly after its initial release, and more in-depth in March 2011, when they also embedded *Netalyzr* on their site among a library of user-friendly testing tools. Until this coverage, the "League of Legends" multi-player online game community referred the largest number of users to us, contributing 19% of *Netalyzr*'s referrals. Their technical sup-port instructions suggest that, if other troubleshooting advice yields

no improvement, users should run Netalyzr and post the results link into their technical support forum where staff and other forum members can diagnose the problem.

We term such usage *third-party diagnosis*: having someone else run *Netalyzr* and simply send along a pointer to the results for more detailed analysis. We specifically designed *Netalyzr* to support this use for debugging the networks of our friends and families. Its persistent use by League of Legends suggests that supporting this mode of operation can help researchers gather longitudinal mea-surements.

The effectiveness of third-party diagnosis in supporting our mea-surement study also rests upon another feature of *Netalyzr*'s design: *bundling*. By this term, we mean that users cannot restrict test ex-ecution to the subset of features they immediately care about, ex-cluding other analyses. For example, the gaming community pri-marily cares for latency, bandwidth, buffering, and loss, and less frequently about DNS resolver correctness. Instead of permitting such partial executions, *Netalyzr* always conducts its entire suite of
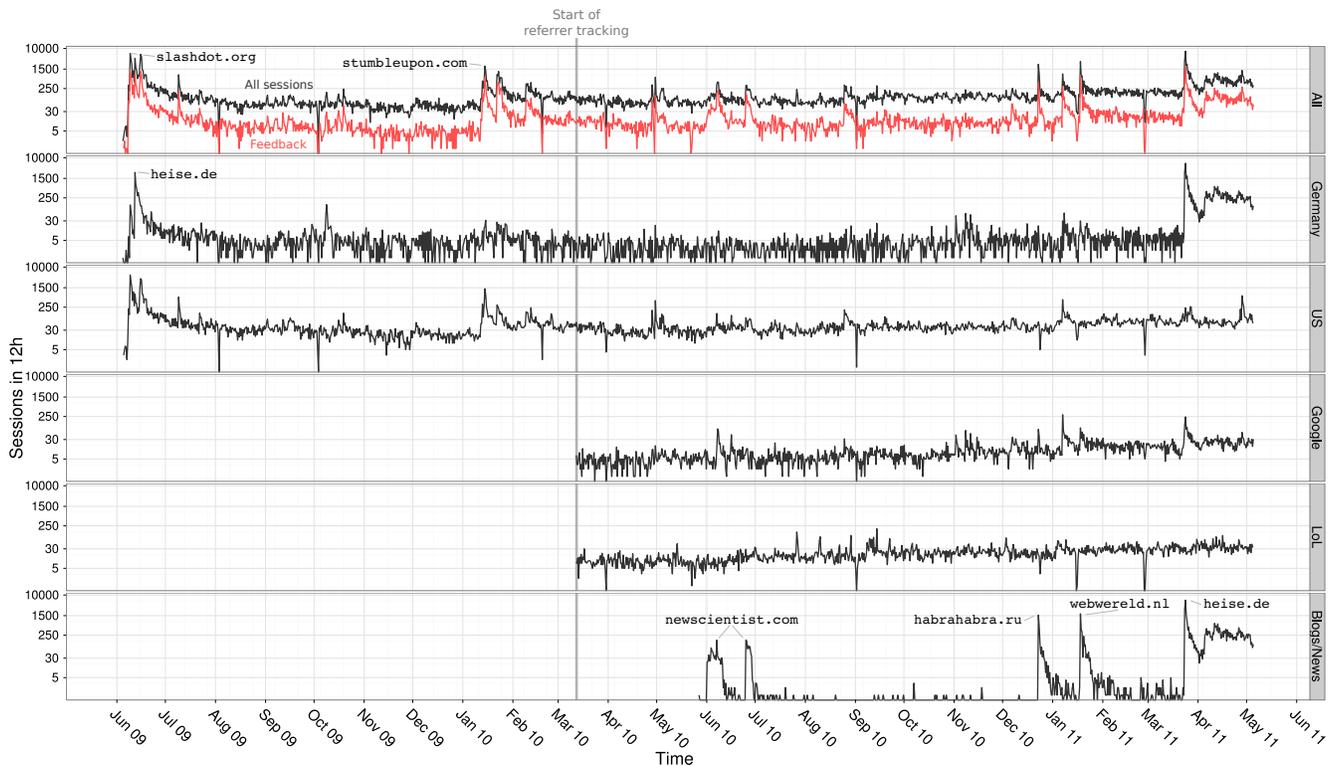
Figure 5: *Netalyzr*'s activity timeline for all sessions, those originating in Germany and the US, and those referred to us by particular websites. Sessions grouped by 12-hour intervals, logarithmic scale.

measurements before providing any results. That we bundle the execution in this manner grants us significant measurement data for network properties of no importance for uses such as debugging performance for gaming.

7% of our referrer-tracked visitors arrive from Google, making it our third-largest traffic originator. Almost all users searched for "Netalyzr", common variations (e.g. "netalyzer", "icsi netalyzr"), and related searches (e.g. "berkeley internet test"). Google automatically corrects the most common spelling defect from "netalyzer" to "netalyzr" in the search results page.

Two IPv6 trials also contributed significant numbers of visitors. Because the operators conducting the trials coordinated with us in advance, these *Netalyzr* runs contain embedded flags that enable the operators to track executions relating to their trials even when the user's browser strips the referral field. 1,743 visitors arrived from Comcast's IPv6 trial, whose IPv6 information portal and trial participant instructions include links to Netalyzr. Another 386 come from an IPv4/IPv6 dual-stack measurement trial conducted by Tore Anderson. In this case, the Netalyzr link only appears when the user's system had an initial problem when attempting to access the dual-stacked test image.

Naturally, referrals such as these bias the data collected, particularly for IPv6 statistics. Consequently, we exclude these tests when calculating IPv6 penetration, problem rates, or other related factors, but use them for the rest of our data analysis.

Netalyzr also receives a significant amount of traffic from occasional flash crowds. heise.de's coverage in March 2011 initiated the largest of these, with 32,000 users to date. The initial coverage of *Netalyzr* on slashdot.org was similar in size, but abated more quickly. In December 2010 a Russian blog sent 2,100 the follow-

ing month, a Dutch technical news blog sent 1,600. Both postings discussed how users could troubleshoot their Internet connection.

Flash crowds are both useful and problematic for measurement studies. The benefit arises from the large number of measurement runs, often from geographically diverse or otherwise interesting areas. However, flash crowds can also result in significantly biased datasets. *Netalyzr* suffers from a heavy "geek bias" from technical flash crowds[1]. Flash crowds also often come at unexpected times, requiring measurement infrastructure capable of handling hundreds of visitors an hour without operator intervention.

Figure 5 shows *Netalyzr*'s user activity timeline. Lack of HTTP referral information before March 2010 limits that part of the data set to total traffic, GeoIP localization, and manually recorded referrers. Google searches and League of Legends create a steady base load that is easy to accommodate given its predictability, although we have seen small flash crowds when League of Legend servers suffer unscheduled downtime. The sharp onsets of flash crowds are harder to handle, naturally, without pre-allocating significant excess capacity.

Note the particularly pronounced spikes of sessions with feedback during the flash crowds: interestingly, during those times up to 44% of users contribute feedback—more than twice the average. The Dutch flash crowd brought such chatty users, the more recent German one somewhat less so, though heise.de's readers were more inclined to share additional technical information.

---

[1] We can observe such biases throughout our data. For example, 8% of *Netalyzr* users use OpenDNS instead of their ISP's resolver in order to conduct DNS lookups.

7. FUTURE WORK

"My computer's clock is not 14 seconds fast, I just happen to live 14 seconds in the future! Watch out for that dog, btw."

—*Netalyzr* user, January 2010

We intend to keep mining future user feedback for usage trends and to inform potential additional tests. For future *Netalyzr* releases, we expect to experiment with refinements to how we solicit feedback, and of what form. One potentially promising approach is to offer the survey during applet execution, a technique inspired by the survey in the HomeNet Profiler [2]. A typical *Netalyzr* run takes four to five minutes to execute, which should leave plenty of time to fill out a more comprehensive survey.

While our users are generally rarely confused by *Netalyzr*'s summary report and also object to its findings only infrequently, requests for additional help are not uncommon (recall Table 1). We are therefore considering ways to incorporate user commentary on our help pages, to allow us to pinpoint which test result explanations in particular we need to make more easily understandable to the non-technical audience.

More generally, the existence of a steady baseline load presents an opportunity to experiment with different forms of feedback solicitation, including randomized controlled experiments, as a means of better understanding the potential of engaging users for network measurement in this fashion.

8. RELATED WORK

"where do i donate?"

—*Netalyzr* user, January 2010

Some network testers aggressively solicit feedback. The FCC consumer broadband test [3] requires that the user provide both their context (home, business size) and address before execution. Ookla's Speedtest.net site [4] contains a small "rate your ISP" selection during test execution and a small optional survey if the user clicks on the "contribute to Net Index" button. The HomeNet Profiler [2] presents a comprehensive survey during application execution.

Other systems also seek user participation, such as the EFF Panopticlick [5] and the Ono plug-in to the Vuze BitTorrent Client [6].

9. CONCLUSIONS

"Thanks a lot, I will send my ISP to hell now."

—*Netalyzr* user, August 2010

Netalyzr's referrer tracking and feedback mechanism provide us with insight into both *Netalyzr*'s utility and technical details regarding the networks we measure. 3% of our users compose free-form responses in order to share a wide variety of information concerning their network connection, with only minor prompting. This information appears to be accurate, as the results match our understanding of different network technology. The reason for using *Netalyzr* also varies considerably as some search out *Netalyzr* specifically, others discover *Netalyzr* as part of a flash crowd, and still others are instructed to use *Netalyzr* to help a third party debug their network. We also note that a "bundling" approach to measurement can lead to gathering data for research of a broader scope than just what the users directly seeks to understand.

10. ACKNOWLEDGMENTS

"YOU GUYS ARE THE BOMBDIGITTY EFFING GENIUSES"

—*Netalyzr* user, March 2011

11. REFERENCES

[1] C. Kreibich, N. Weaver, B. Nechaev, and V. Paxson, "Netalyzr: Illuminating the edge network," in *Proceedings of the ACM Internet Measurement Conference (IMC)*, Melbourne, Australia, 2010.

[2] L. D. Cioccio, R. Teixeira, and C. Rosenberg, "HomeNet Profiler," http://cmon.lip6.fr/hnp/pages/home.

[3] "FCC Consumer Broadband Test," http://www.broadband.gov/qualitytest/.

[4] "Ookla http://speedtest.net."

[5] P. Eckersley, "How Unique Is Your Web Browser?" in *Proceedings of Privacy Enhancing Technologies Symposium (PETS)*, 2010.

[6] D. Choffnes and F. Bustamante, "Taming the Torrent: A practical approach to reducing cross-ISP traffic in P2P systems," in *Proceedings of ACM SIGCOMM*, 2008.