NORTHWESTERN
UNIVERSITY

Electrical Engineering and Computer Science Department

Technical Report
NWU-EECS-10-04
March 2, 2010

Network Monitoring is People: Understanding End-user Perception of Network Problems

J. Scott Miller **Amit Mondal** **Rahul Potharaju**
Peter A. Dinda **Aleksandar Kuzmanovic**

Abstract

It is widely assumed that certain network characteristics cause end-user irritation with network performance. These assumptions then drive the selection of quality of service parameters or the goals of adaptive systems. We have developed a methodology and toolchain, SoylentLogger, that employs user studies to empirically investigate such assumptions. SoylentLogger collects client-centric network measurement data that is labeled by the end-user as being associated with irritation at perceived network performance (or not). The data collection and labeling occurs in real-time as the user normally uses the network. We conducted a user study that produced the equivalent of two years of labeled data, and then used that data to test various assumptions about network sources of user irritation that are commonly made. A number of these assumptions were found wanting.

This project is made possible by support from the National Science Foundation (NSF) via grants 0720691 and 0746360

Keywords: network monitoring, human factors, empathic systems

Network Monitoring is People: Understanding End-user Perception of Network Problems

J. Scott Miller Amit Mondal Rahul Potharaju
Peter A. Dinda Aleksandar Kuzmanovic
{jeffrey-miller,a-mondal,rahul,pdinda,akuzma}@cs.northwestern.edu
Northwestern University

***Abstract*—It is widely assumed that certain network characteristics cause end-user irritation with network performance. These assumptions then drive the selection of quality of service parameters or the goals of adaptive systems. We have developed a methodology and toolchain, SoylentLogger, that employs user studies to empirically investigate such assumptions. SoylentLogger collects client-centric network measurement data that is *labeled* by the end-user as being associated with irritation at perceived network performance (or not). The data collection and labeling occurs in real-time as the user normally uses the network. We conducted a user study that produced the equivalent of two years of labeled data, and then used that data to test various assumptions about network sources of user irritation that are commonly made. A number of these assumptions were found wanting.**

I. Introduction

End-users increasingly experience changes in application or service performance due to fluctuations in the underlying network characteristics. This is due to popularity of interactive network applications and services, the expansion of connectivity, and the increasing complexity of the Internet. To maintain a high level of end-user satisfaction, significant research has been done into network control systems, such as quality of service systems and adaptive systems [17], [19], [12], [8], and on modeling user reaction to and satisfaction with specific applications (*e.g*, VoIP [13]). Broadly speaking, such systems attempt to enforce network characteristics within a range such that the user is not irritated by the network.

This raises the natural question: *What network characteristics lead to irritation?* There is surprisingly little empirical work that addresses this question, though recent work [7] has suggested that user irritation is frequent and pervasive. One approach is to find application-level metrics desirable for avoiding irritation and then transform these metrics to desirable network characteristics that can be demanded of a network control system. However, in many cases, application-level empirical data is also rare, and, in others, the transformation may be quite difficult. Finally per-flow or per-flow class control is generally unavailable. A second approach is essentially based on assumptions, *rules of thumb* for what network characteristics lead to irritation. These assumptions can then be built into network control systems. In this paper, we empirically study several such rules of thumb; *more generally, we consider the correlation of user irritation and network characteristics.*

Our approach was to conduct a careful user study "in the wild." The study produced client-centric network measurement data that was *labeled* as being irritating or not. These labels were applied in real-time by the participants in our study while they used the network as they normally would. Each participant in our study ran our SoylentLogger software on their Windows computer for at least three weeks. As described in more detail in Section II, SoylentLogger periodically probed network conditions. The participant could at any time express irritation with (perceived) network performance by pressing a labeled button. This form of feedback, and its analysis, was pioneered by us to study user irritation with restrictions on the availability of computer resources such as CPU bandwidth, physical memory pages, and disk bandwidth [4], and has more recently employed to understand the consequences of power management decisions that lower processor frequency [11]. In the networking domain, a related approach to inexpensively measure mean opinion scores for audio and video QoS metrics has recently been described [1]. We describe the design of SoylentLogger and of our user study in detail in Section III.

The labeled network data allows us to test whether the assumptions or rules of thumb hold. For a given assumption, we can create a hypothesis and conduct a hypothesis test via a query over the assembled labeled data. For example, we can find cases that exhibit network conditions corresponding to an assumption and then test if there is any difference in the probability of irritation for those cases and others. Conversely, we can separate the network data that is labeled as being irritating from that which is not and then search for network characteristics that actually differ between the two groups. Section IV tests a variety of assumptions/rules of thumb based on this approach.

The contributions of this paper are the following.

- We advocate the study of the effects of network characteristics on *measured* end-user irritation.
- We describe a methodology and a toolchain for carrying out such studies.
- We collect what we believe is the first set of network data labeled with end-user irritation information. We will make this data available in an anonymized form for others to use.
- We tested a range of assumptions or rules of thumb about how network characteristics affect user irritation.

It is important to point out that the assumptions we test in this paper are a selection of what may be possible given the data we have collected. Nonetheless, the results are often surprising, as described below.

1) End-users are generally quite good at correctly attributing bad application performance to network conditions, when that is the case.
2) It is widely believed that the performance of small connections is most critical to the end-user experience. We found that the median size of connections associated with irritation was 2.8 times larger than those not associated with irritation, while the median duration was 34.6 times higher. *Lethargic mice* seem responsible for a significant fraction of end-user irritation with the network.
3) Both applications and services vary considerably in the level of irritation that their users see per byte transferred, or connection made. For example, the three ASes with the highest irritation rates represent $< 5.1\%$ of observed traffic, but 48.9% of all traffic associated with irritation.
4) User irritation is stateful. Once a user has become irritated, that irritation is likely to persist.
5) Neither the Windows-based "Signal Quality" metric, nor the wireless NIC-based RSSI metric are good predictors of user irritation, although the former is slightly better.
6) On wireless networks, the rate of irritation depends strongly on the choice of access point.

Our empirical results suggest that a large fraction of user irritation is associated with a small fraction of the possible sources. This implies that end-users, given a low-overhead feedback mechanism, could focus the maintenance of network components, for example access points and CDNs, on those that would provide the maximum user-perceived benefit per dollar spent.

II. SoylentLogger

In order to collect data which can be used to test networking assumptions and rules of thumb, we developed SoylentLogger, a tool allowing end-users to indicate when they are dissatisfied with their network service. The tool simultaneously performs a set of network and local measurements at regular intervals and in response to user irritation events. SoylentLogger gives us a rare look into the experiences of the end-user in real-world environments without degrading that user's performance.

A. Design

We developed SoylentLogger for Windows XP and Vista on the .NET 2.0 platform to maximize the number of participants eligible for our study. While we tested SoylentLogger extensively both within our research group and with a larger group of "beta" users, we still needed a mechanism to allow us to correct undesirable application behavior or implement and quickly deploy changes to the data collection methodology. Requiring user intervention would needlessly delay or prevent necessary changes to the instrumentation, reducing the already limited time we have with each user. Thus, SoylentLogger consists of two components: a bare minimum framework and an assembly that can be easily updated.

The framework runs as Windows service responsible for monitoring the status of the dynamically-loaded assembly and initiating an upgrade when the new version becomes available. The service component of SoylentLogger periodically checks for new updates at a pre-specified web location, downloading and loading a new assembly once a newer version is available. This provides us the control to automatically update the SoylentLogger application at participants' machines if we need to improve our logic at later point of time.

SoylentLogger periodically uploads data when the size of the local log reaches a fixed size, a size small enough such that data is uploaded every few minutes. This gives us immediate feedback that is critical for adjusting the software and collection methodology while the study is active. As we will show in Section III, the overhead of this communication is small.

We also provide the participants with the ability to stop and restart the service at any point of time. SoylentLogger creates a "system tray" icon using which the users can easily control the SoylentLogger application. In practice, however, we found that users rarely disabled the application.

B. Irritation

SoylentLogger allows users to express their irritation using a globally mapped key (F8) or a "system tray" menu item. We refer to a user's expression of irritation as an "irritation event." When an irritation event occurs, SoylentLogger immediately collects data that would otherwise be collected periodically. This allows us to capture some transient events that might go undetected by the periodic monitoring.

C. Metrics

The data collected by SoylentLogger consists of two components: 1) Local software and hardware factors and 2) network factors collected from the end-host's vantage point. While the latter is used to understand the network dynamics encountered by the end-user (in the context of user-reported irritation and otherwise), the software and hardware diagnostics are used to determine any factors that the user attributes to the network state that are not related to this root cause. We describe all of these factors in Figure 1.

All logged data is written to a text file that is periodically compressed and uploaded to a secure server.

Hardware and software factors: We instrumented SoylentLogger to measure a number of factors unrelated to the network to help characterize the rate at which users falsely attribute machine-local factors to the network. Furthermore, we also collect information about the user's current "task" and attentiveness to help guide our exploration of network factors. Each factor is collected at a fixed rate and many are collected in response to user irritation.

To accurately compare conditions when the user expresses irritation and when the user is *physically present* but not

Factor	Collection Period	Details
CPU Utilization	1 second	Aggregate system load.
Process Statistics	30 seconds	Per-process performance and resource utilization data including cumulative time in execution, working memory size, and number of page faults.
Application Focus	10 seconds	The name and title of the current window in focus.
User Activity	30 seconds	Total mouse movement, button clicks, and keystrokes in the last interval.
Offered Throughput	5 seconds	The average offered throughput of each netflow during the collection period.
Application RTT	Continuous	The elapsed time between the transmission of a TCP packet and the arrival of the acknowledging packet.
Receiver Signaling	Continuous	TCP packets for which the receiver window is set to zero, potentially indicating load on a remote server.
Duplicate Packets	Continuous	TCP packets repeated by either the sender or receiver.
Web Traffic	Continuous	The URL and method of each HTTP request.
Link	5 seconds	The properties of the current network interface being used, such as IP and link speed.
Wireless Interface	5 seconds	Additional information on the wireless link including base station MAC address and signal strength.

Fig. 1. The hardware, OS, network, and user metrics logged by SoylentLogger along with the logging period.

irritated, we need to determine when the user is actively engaged with their computer. Without doing this, any potentially disruptive network activity experienced while the user is away from their computer would be falsely characterized as *non-disruptive*. To determine the user's presence, we periodically collect the mouse movement, click rate, and typing rate. While this simple methodology will sometimes lead us to consider an actively engaged user as absent, such as while the user is watching a long, full-screen video, we found it to be the best approach to minimize both error and the involvement of the user.

Network factors: SoylentLogger uses the WinPcap library [16] to capture all outgoing and incoming packets from the user's network interface. We gather the connection start time, establishment time, and completion time in addition to the number of bytes and packets traversed in both directions and the four-tuple associated with each connection. We also collect instantaneous round-trip-time, throughput, loss rate of each connection. We further tag each TCP and UDP socket with its parent application. In addition to the metrics described in Figure 1, SoylentLogger periodically initiates ping and traceroute probes to the 25 most frequently accessed IP addresses to assess the end- and remote-host's connectivity.

III. User study

To collect the data necessary to investigate network sources of user irritation, we conducted a rigorous user study.[1] 32 users used the SoylentLogger software over a period of three weeks, yielding 899 irritation events and 20.0 GB of data. This amounts to roughly 2 years worth of irritation-annotated trace data. Data gathered by SoylentLogger indicates that the overhead of the software was sufficiently low that the presence of the SoylentLogger did not add additional irritation events.

A. Design

We recruited 32 participants from a pool of students, staff, and faculty at Northwestern University using a combination of flier and email advertising. The study group, primarily undergraduate students, received network connectivity via both the university and a number of regional ISPs.

After signing up for our study, participants visited our lab to have the software installed. After signing a consent form for our study,[2] each filled out a background questionnaire. We found that our subjects use a variety of network services and that few have academic experience in CS, CE, or EE.

Once the background questionnaire was completed, the participants read an overview of the study that included the following instructions: "We ask that you press (the irritation button) when you are uncomfortable or dissatisfied with the network service being provided to the applications you are using." The document also gave several examples of network performance issues ranging from slow-loading web content to interrupted video and audio streaming. Our documents stress that the users are to signal irritation with the *performance* of the network applications being used and not the *content* of those applications.

Once the participant completed the forms and read the study documents, the investigator installed the SoylentLogger software. Installing the software in-lab allowed us to verify that the software was correctly installed as well as answer any questions participants had about the study.

While our study was limited to only three weeks, we needed a mechanism to remind participants to indicate any network dissatisfaction using the tool. Without such a reminder, the data would be subject to a bias where irritation events would be less frequent later in the study. We initially considered having the software periodically issue a direct notification to the user. This was deemed inappropriate as it is likely to bias the participants towards generating irritation events shortly after being prompted. We also considered the use of reminders sent via e-mail or text message, though these mechanisms are likely to have the same effect. Finally, we decided on placing a small sticker on each participant's laptop at a location visible during normal usage. The sticker read "Press F8 when irritated with the network" and remained on each participant's computer for the duration of the study.

[1]The materials used during the study can be found at http://empathicsystems.org/SoylentLogger

[2]The study and related documents were approved by our Northwestern University's Institutional Review Board

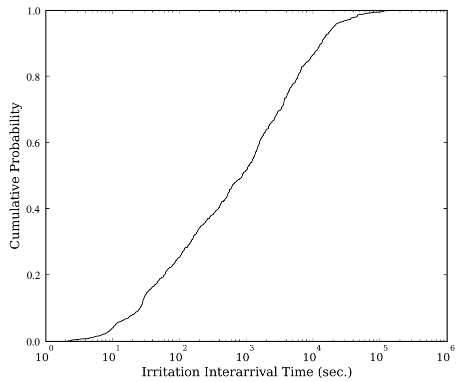


Fig. 3. Interarrival times of user irritation events. The interarrival time only includes times that the user is actively using their computer. Times when the user is absent are discarded.

After completing the study, each participant was paid \$25 for his or her time.

B. SoylentLogger overhead

The study did not alter the network service that each participant received and was merely intended to characterize how end-users perceive their network service. Because SoylentLogger collects a substantial amount of data through a variety of means, including deep packet inspection, it is possible that the additional overhead of the tool may cause users to report irritation. There is thus a tradeoff between the resolution of data that we are able to collect and the risk of interfering with the participants' experiences. During our testing of SoylentLogger, we tuned the activity of the tool to keep resource utilization within a reasonable range.

In Figures 2(a–c) we report the network, memory, and CPU utilization of SoylentLogger during our study. The 95^{th} percentile network overhead, which includes uploading study data and checking for updates, is less than 10 kBps in both the upstream and downstream directions. The memory overhead is similarly insignificant as most participants have several gigabytes of physical memory. While the CPU utilization of our tool can, at times, be significant, as we will show in Section IV, there is little correlation between CPU utilization and user irritation. Both the 95^{th} percentile memory and CPU overhead are within the comfort ranges reported in [4].

C. Data collected

In total, the participants in our study generated 899 irritation events, averaging over 1.2 events per user per day. Figure 3 shows the cumulative distribution of irritation interarrival times for each user. In calculating interarrival times, we only consider the time between irritation events when the user is "active" according to the methodology described in section IV-A. A small number of our irritation events are closely spaced, implying that some users pressed the irritation key multiple times in quick succession. However, over 95% of the irritation events occur more than 10 seconds after the previous event, implying frequent presses are not common.

From this data, it is clear that end-users are frequently dissatisfied with their network usage—50% of irritation events occur within 17 minutes of a previous irritation event. In the next section, we will analyze how this data confirms or refutes several hypotheses about the user experience.

IV. Hypotheses

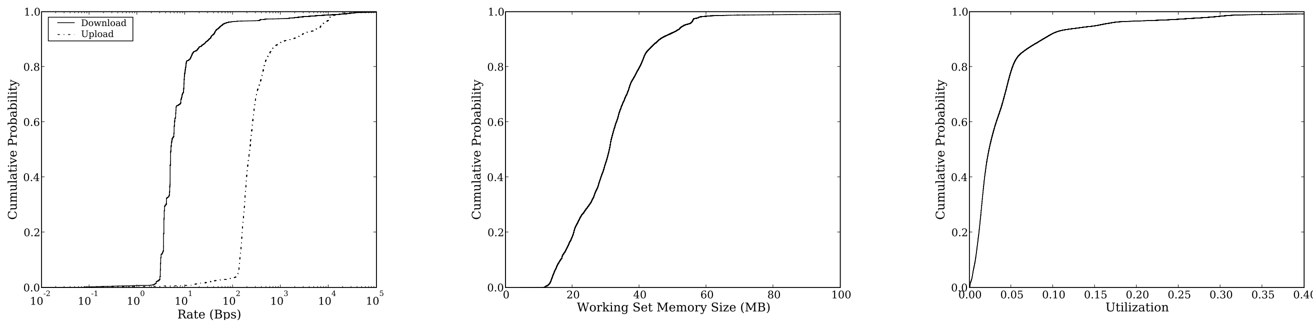
We now employ the data from the user study to test hypotheses or rules of thumb about the sources of user irritation with the network.

A. Methodology

In evaluating hypotheses we use a common methodology for correlating factors with irritation events. If a factor correlates with irritation, we assume it changes near the irritation event. To make the meaning of "near" precise, we define two parameters that together specify a window of time around an irritation event in which the factor may change and be considered correlated with the event. The window is ω seconds long and is displaced from the irritation event by an interval τ. ω can be thought of as how long the change must persist to cause irritation, while τ is the delay between the cause and the user's reaction. For the data presented here, $\tau = 1$ second and ω is varied as discussed in the text. ω and τ define the "irritation window." Data outside of the window is assumed to be unrelated to the irritation event.

Our methodology must also consider the user's presence, as failing to account for times when the user is absent would introduce two biases. First, anomalous activity that would otherwise cause irritation would be unnoticed. Second, data associated with absence, such as low CPU utilization, would erroneously become associated with a lack of irritation. To eliminate these biases, we only analyze log data for which we know the user is present, based on the keyboard and mouse activity we recorded. We filter out data logged outside of a 60 second window centered around each instance of user activity.

For many of the hypotheses we consider, we compare the distribution of a given factor outside of all irritation windows with the distribution of those occurring within irritation windows. We evaluated the sensitivity of our results to the choice of ω and τ. For ω, we show the results in the paper. For space reasons, and to avoid making our figures unreadable, we omit the results of sensitivity to τ; the sensitivity to τ is much lower than the sensitivity to ω within a reasonable range. As an example, consider Figure 4(c), which compares the CDF of aggregate netflow throughput outside of all irritation windows ("No Irritation") with the CDFs for those in irritation windows, for a range of ω sizes (*e.g.*, "Irritation (Window 5 sec.)"). We can see that the median aggregate throughput is roughly an order of magnitude greater during user irritation events and that this difference is most pronounced when ω= 2 seconds.



(a) Network Overhead (b) Memory Overhead (c) CPU Overhead

Fig. 2. Overhead of running the SoylentLogger tool.

B. Results

Hypothesis 1: Users can distinguish between local and network sources of irritation.
Result: Supported by our evidence.

It is critical to our analysis that users be able to differentiate irritation that is due largely to the network from irritation whose genesis is elsewhere. In other words, we need to demonstrate that users meaningfully assign blame to the network. While our study documents informed participants that they were only to indicate irritation with the performance of network services, it is possible that local conditions unrelated to the network, such as application load, would falsely trigger an irritation event. To evaluate the effects of local conditions, we consider how well two local factors indirectly related to network performance are correlated with user-reported irritation: CPU utilization and page fault rate. Overall, we find that both factors are weakly correlated with user irritation.

Figures 4(a-b) show the CPU utilization and page fault rate for a range of ω values. Both distributions show little difference in the presence of an irritation event. In contrast to this, Figure 4(c) shows that the distribution of link utilization is noticeably different, with the median throughput almost an order of magnitude greater during irritation. Indeed, greater network activity in the form of increased throughput implies a higher rate of user irritation. Thus, we conclude that it is unlikely that many cases of irritation are due to application load or memory contention and that users can successfully distinguish between local and network sources of irritation.

Hypothesis 2: Most user irritation is associated with small connections.
Result: Supported by our evidence, with further observations.

It is widely assumed that small flows are critical to the end-user experience and that the poor performance of small flows dominantly affects users' perception of the network service. As a result, the performance of small flows has traditionally been one of the key QoS metrics [9], [2], [10]. In order to improve the performance of small flows, researchers proposed adaptive bandwidth allocation schemes that aim to minimize file-transmission times using filesize-based service differentiation. As an example, Guo and Matta [3] use RIO in core routers and a packet classifier at the edge to distinguish

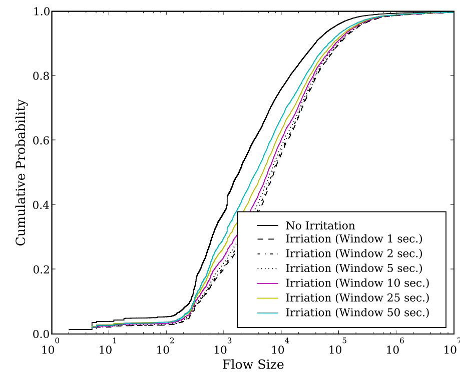


Fig. 5. Flow size distribution for a range of window sizes. Irritation events are associated with larger flows, on average, than flows not associated with irritation. However, the absolute size of these flows is not dramatically different.

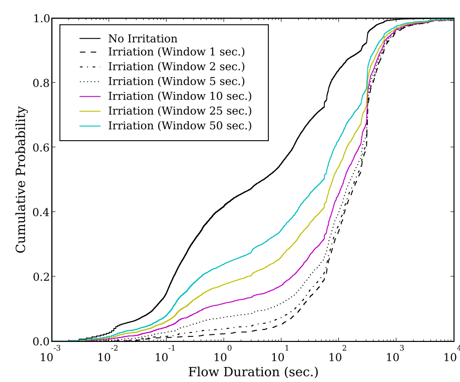
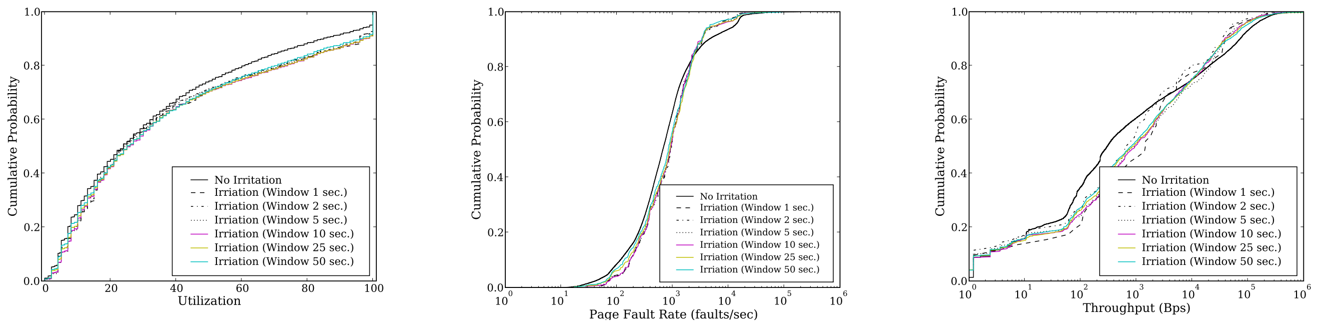


Fig. 6. Flow duration distribution for a range of irritation window sizes. When considering window sizes less than 10 seconds, the distribution of flow durations is substantially different during irritation events.

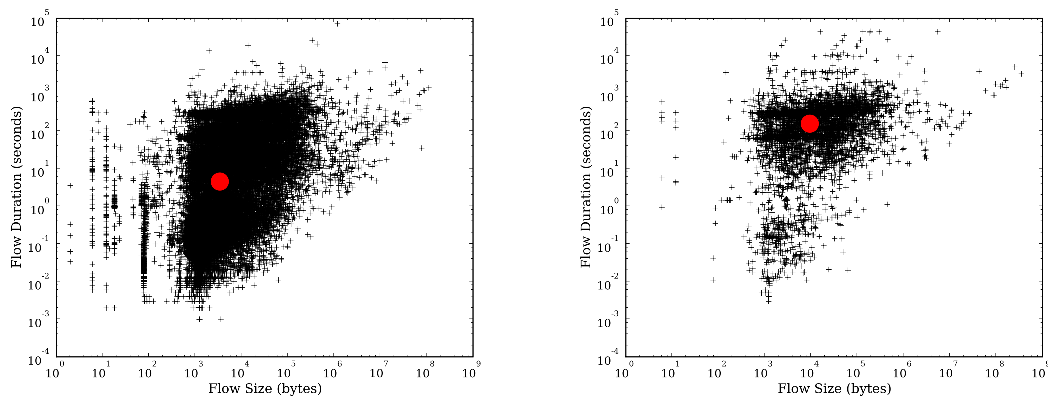
between large and small TCP flows. Yang and de Veciana [18] develop TCP/SAReno in which the AIMD parameters dynamically depend on the remaining file size.

Overall, we find that while connections present during irritation tend to skew larger in size and longer in duration, the majority of the connections associated with irritation are quite small. In Figure 5, we compare the distributions of flow sizes both during irritation and not. The median flow size is



(a) CPU Utilization (b) Page Fault Rate (c) Aggregate Network Throughput

Fig. 4. Correlation with machine-local conditions and network utilization. While the CPU utilization and page fault rate show little correlation with with irritation, the aggregate network throughput is shows noticeable difference during user irritation.



Flows outside of irritation. Flows in irritation (ω = 5 seconds).

Fig. 7. Flow duration and total data transferred for both netflows associated with irritation and not. The circle on each plot is centered at the median flow size and duration. While the flows present during user irritation tend to be longer in duration, the size of those flows is comparable to those apart from irritation.

2.8 times larger during irritation, although the absolute size of these these flows is still less than 10 KB. As shown in Figure 6, the flow duration during irritation is considerable longer, with the median duration 34.6 times larger during irritation. Figure 7 provides another way of looking at this result. Figure 7(a) plots flow duration versus flow size for each flow not associated with irritation, while Figure 7(b) does the same for those flows that are associated with irritation. The circle in each plot represents the median flow size and duration. User irritation is most closely associated with small flows that are long-lived, which might be termed the *lethargic mice.*

Hypothesis 3: User irritation is dependent on the application and services with which that user interacts.
Result: Supported by our evidence.

We suspected that the network applications a user interacts with will vary in their association with irritation. This is a natural assumption as applications vary in their QoS requirements, resulting in some being more sensitive to disruptions in service. We also suspected that the content provider also plays a role in user irritation. For example, to enhance web browsing experiences, content distribution networks (CDNs) move web content closer to clients by caching copies of web (and other) objects on thousands of servers worldwide. It has

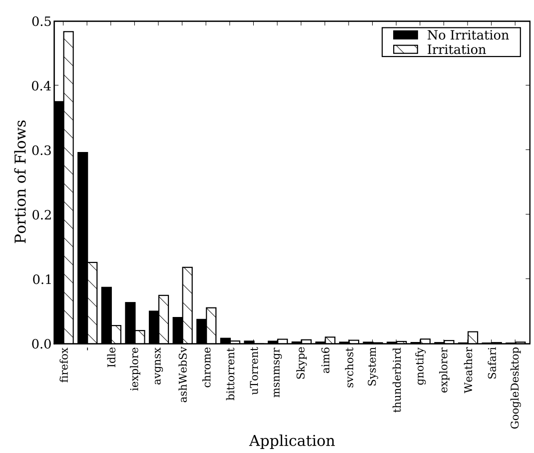


Fig. 8. The portion of non-irritation and irritation traffic associated with each application. Firefox, Chrome, and the Avast Internet Security Suite (ashWebSV) are associated a higher proportional of netflows during irritation than not, while idle system activity is less likely to associated with irritation.

been demonstrated that this approach can help improve Web response times (*e.g.*, [6], [14]) and so it is assumed that CDNs positively impact users' perceived QoS. We find that this is not always the case as the rate of user irritation associated with

Host	No Irritation		Irritation		Total	% Bytes in
	Traffic (MB)	Flows	Traffic (MB)	Flows	Traffic (MB)	Irritation
Google Inc.	8402.24	85133	295.85	1376	8698.08	3.40
Comcast Cable Communications Inc.	6475.09	7084	0.04	3	6475.13	< 0.01
Northwestern University	4242.61	88970	66.10	908	4308.71	1.53
Level 3 Communications	3988.20	18024	234.10	582	4222.30	5.54
Limelight Networks Inc.	3155.00	14608	2.51	110	3157.51	0.08

(a) Top-5 ASs by traffic volume.

Host	No Irritation		Irritation		Total	% Bytes in
	Traffic (MB)	Flows	Traffic (MB)	Flows	Traffic (MB)	Irritation
Advanced Video Communications Inc.	767.08	3032	452.35	10	1219.43	37.10
Global Crossing Ltd.	480.51	1325	240.20	19	720.71	33.33
NTT America Inc.	559.78	5379	246.13	45	805.91	30.54

(b) Top-3 ASs by irritation rate.

Fig. 10. Traffic quantity and irritation rates for a selection of ASs. ASs with seemingly identical responsibilities, such as content replication and delivery, show considerable variation in their respective irritation rates. Also, not that a small number of ASs are associated with large amounts of irritation.

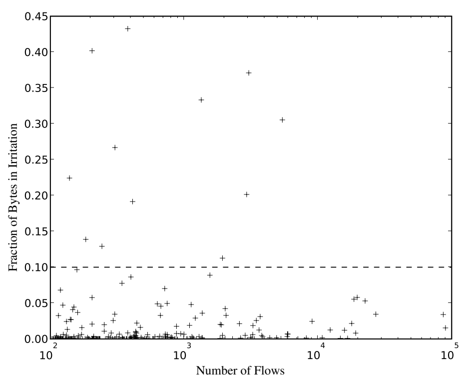


Fig. 9. For each destination AS, the number of flows seen to the AS is plotted versus the fraction of bytes associated with irritation events. Note that even for destinations for which we have considerable data, the rate of irritation can be very high.

CDN traffic is highly variable across providers.

In Figure 8, we plot the percentage of flows attributed to each application, along with the percentage of flows associated with irritation attributed to each application. A difference between the two bars for a given application implies that the application was associated with a disproportionate amount of irritation. Almost 40% of the netflows seen in our study are generated from Firefox, and 75.8% of the flows in our study come from web traffic. Surprisingly, we find that Internet Explorer has a lower rate of irritation as compared the other browsers. We hypothesize that this is due to participants in our study using different browsers for different sites and services. As we will show, the choice of service plays an important role in user irritation. Another 30% of the flows terminated before SoylentLogger could query the operating system for the associated application, giving the "-" label in the figure. Consistent with our earlier findings, these short-lived connections are less likely to be associated with irritation.

The volume of web traffic makes it clear that it is not sufficient to consider only the application associated with each netflow. To determine the web application in use, we first attempted to reverse map each IP address using DNS. Unfortunately, this method covered little more than 20% of our data. Next, we considered using the IP to Autonomous System Number (ASN) maps provided by Cymru [15]. Using this data we were able to map over 96% of our netflows.

To compare the amount of irritation associated with each autonomous system (AS), we consider the amount of data associated with irritation. As before, we associate a flow with an irritation event if that flow is present during the irritation window specified by that event. If a flow is associated with irritation, we consider *all* bytes of that flow as irritation bytes. Thus, the irritation rate is the number of bytes in all such flows divided by the total number of bytes transferred to or from that AS. While this methodology places a greater weight on larger flows, we feel that it appropriately compares the consumption-based pricing model most content hosts adopt.

In Figure 9 we plot, for each of the 284 ASs that have over 100 flows, the number of flows seen with that AS as a *destination* and the fraction of all bytes transferred to or from that AS that are associated with an irritation event. We find that there are a large number of ASs with substantial per-byte irritation rates. 12 of these ASs have over 10% of their traffic associated with irritation, which implies a disproportionate amount of user irritation.

In Figure 10(a-b) we show a traffic and irritation summary for several high-activity and high-irritation ASs. Figure 10(a) shows the top 5 ASs in terms of traffic. There is both considerable data for all of these providers and visible stratification among them. For example, while Level 3 Communications and Limelight Networks provide similar content delivery services, the two have very different irritation rates, with Level 3's irritation rate being 69 times greater.

Figure 10(b) shows the top 3 ASs (of those with more than 1000 flows) in terms of irritation rate. These hosts show very high rates of irritation emanating from a small number of large flows. While these ASs represent less than 5.1% of traffic, they make of 48.9% of all bytes associated with irritation.

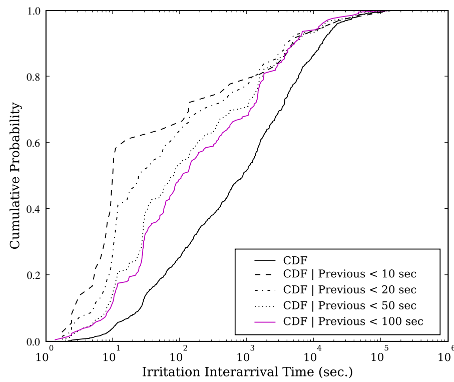


Fig. 11. The distribution of per-user irritation inter-arrival times given that the previous interarrival time is less than some threshold.

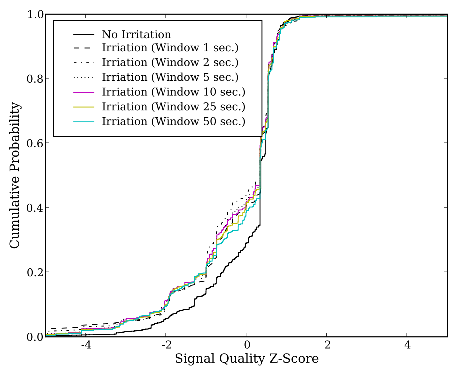


Fig. 12. Windows "Signal Quality" metric as we sweep though a range of window sizes. The greatest difference between the two distributions is reached with a window size of 2.

Hypothesis 4: User irritation is stateful.
Result: Supported by our evidence.

To characterize the extent to which user irritation is stateful, we consider triples of sequential irritation events. We plot the distribution of the inter-arrival time between the second and third events given that the inter-arrival time of the first and second is below some threshold. We then vary this threshold for a range of values. If user irritation is indeed stateless, then these distributions should not differ.

Figure 11 plots the distributions. It is clear that irritation rate is indeed influenced by the prior knowledge of the user's irritation rate, implying that user irritation is stateful. When a user expressed irritation twice in the preceding 10 seconds, the next irritation event is likely to be generated within 10 seconds in 60% of the cases. This implies that irritation, once caused, tends to persist for the user.

Hypothesis 5: RSSI and link quality indicators predict user irritation on wireless networks.
Result: Not supported by our evidence.

We next explore the correlation between irritation events generated during the study and the condition of the local wireless network. To do this, we consider two metrics: the "Signal Quality" metric provided by Windows and the vendor-

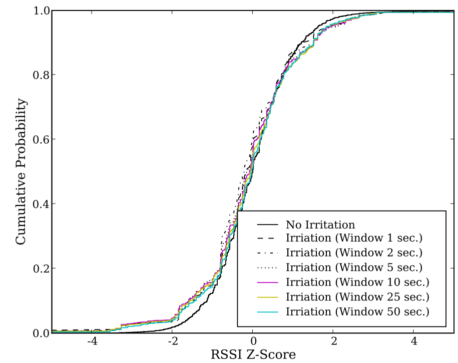


Fig. 13. The distribution of normalized received signal strength reported by the wireless interface during irritation. While there are some irritation events associated with lower signal strength, these represent less than 20% of irritation events at wireless access points.

specific received signal strength indicator (RSSI) exposed by the 802.11 hardware. Overall we find that irritation events are most correlated with the "Signal Quality" metric and that disruptions are not transient.

During our testing of the SoylentLogger tool, we noticed that the range of RSSI values varied depending on the user, implying that the value for this parameter was dependent on the wireless adapter in the user's hardware. This was confirmed by examining the 802.11 specification, which specifically leaves units of the RSSI value undefined [5]. Before we could effectively compare RSSI values between users, we needed some way to normalize their values. Because we are interested in examining when either metric is deviating from a normal range, we decided to transform the raw value into a z-score normalized for each user.

In Figure 12 we plot the distribution of the normalized Windows "Signal Quality" metric both in the presence of irritation and not for a range of ω. The difference between the distributions is maximized when $\omega = 2$ two seconds, though the signal quality metric is not sensitive to ω. This, along with the fact that signal quality is consistently lower near irritation events, implies that while users are more likely to express irritation if signal quality is decreased, it does not appear that this is caused by transient physical-layer disruptions.

In Figure 13 we plot a distribution of normalized RSSI values around irritation along with a baseline. We see the largest difference when the RSSI value is more than one standard deviation below the average RSSI value for that hardware. However, fewer than 20% of the measured RSSI values during irritation events fall into this category. Once again, we see little sensitivity to the choice of ω value, implying that the irritation is unlikely to be caused by transient changes in signal strength.

Overall, the "Signal Quality" metric appears to have a stronger correlation with user irritation as compared to the RSSI measure. However, when we consider how well signal quality *predicts* irritation, the measure does little good. In Figure 14 we show the false-positive and false-negative rate for

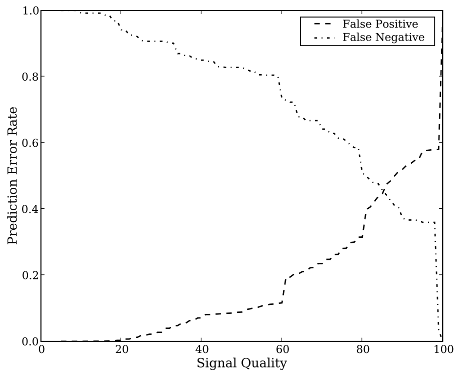


Fig. 14. The predictive power of the Windows "Signal Quality" for user irritation, using a threshold-based predictor, as a function of the threshold. There is no threshold which provides low false negative and false positive rates simultaneously.

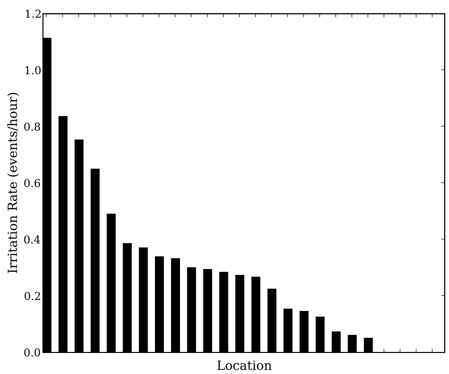


Fig. 15. The rate of irritation events for the top-25 most frequently visited access points, sorted in order of decreasing irritation rate.

a threshold-based predictor using various signal quality values. While important, signal quality is not a strong predictor of user irritation by itself.

Hypothesis 6: User irritation is affected by user location.
Result: Supported by our evidence.

Finally, we consider the extent to which irritation is associated with wireless access points. Figure 15 shows the rate of irritation for the 25 most frequently visited access points, each having at least 5 hours of user activity. If each access point were equally likely to be associated with user irritation, we would expect a uniform distribution; however, this is not the case. Also, across all access points for which we have more than 1 hour of trace data, the top 20% of locations in terms of irritation rate are responsible for 64% of the overall irritation rate. Improving service at a small subset of locations may result in a disproportionate reduction in total irritation.

V. Conclusion

We presented a tool and a methodology for collecting and studying end-user irritation with the network "in the wild." We used the data we collected from an extensive user study to test a range of assumptions or rules of thumb that are commonly made in network control systems or adaptive applications. The most important implications of our work so far are that users are able to appropriately assign blame to the network when they are irritated, and that a small number of sources seem to disproportionately contribute to the irritation experienced by those users. This suggests that low-overhead end-user feedback, such as in our tool, could be used to focus the resources of network maintenance on these "hot spots", maximizing return on investment. At the present time, we are testing further hypotheses using our data, and we are investigating how a network irritation hot spot detection system might be designed.

References

[1] K.-T. Chen, C. C. Tu, and W.-C. Xiao. OneClick: A framework for measuring network quality of experience. In *Proceedings of IEEE INFOCOM 2009*, 2009.

[2] M. Christiansen, K. Jeffay, D. Ott, and F. Smith. Tuning RED for Web traffic. *IEEE/ACM Transactions on Networking*, 9(3):249–264, 2001.

[3] L. Guo and I. Matta. The war between mice and elephants. In *Proceedings of IEEE ICNP '01*, Riverside, CA, Nov. 2001.

[4] A. Gupta, B. Lin, and P. A. Dinda. Measuring and understanding user comfort with resource borrowing. In *HPDC '04: Proceedings of the 13th IEEE International Symposium on High Performance Distributed Computing*, pages 214–224, Washington, DC, USA, 2004. IEEE Computer Society.

[5] IEEE Computer Society. Wireless LAN medium access control (MAC) and physical layer (PHY) specifications, June 2007. IEEE Standard 802.11, 2007 Edition.

[6] J. Kangasharju, K. Ross, and J. Roberts. Performance evaluation of redirection schemes in content distribution networks. *Computer Communications*, 24(2):207–14, Feb. 2001.

[7] T. Karagiannis, E. Athanasopoulos, C. Gkantsidis, and P. Key. HomeMaestro: Order from chaos in home networks. Technical Report MSR-TR-2008-84, Microsoft Research, 2008.

[8] J. Kephart and D. Chess. The vision of autonomic computing. *Computer*, 36(1), January 2003.

[9] A. Kuzmanovic. The power of explicit congestion notification. In *Proceedings of ACM SIGCOMM '05*, Philadelphia, PA, Aug. 2005.

[10] L. Le, J. Aikat, K. Jeffay, and F. Smith. The effects of active queue management on Web performance. In *Proceedings of ACM SIGCOMM '03*, Karlsruhe, Germany, Aug. 2003.

[11] B. Lin, A. Mallik, P. Dinda, G. Memik, and R. Dick. User- and process-driven dynamic voltage and frequency scaling. In *Proceedings of the 2009 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS)*, April 2009. Summary published in SIGMETRICS 2007.

[12] B. D. Noble, M. Satyanarayanan, D. Narayanan, J. E. Tilton, J. Flinn, and K. R. Walker. Agile application-aware adaptation for mobility. In *Proceedings of the 16th ACM Symposium on Operating Systems Principles (SOSP)*, 1997.

[13] A. Rix. Perceptual speech quality assessment - a review. In *Acoustics, Speech, and Signal Processing, 2004. Proceedings. (ICASSP '04). IEEE International Conference on*, volume 3, pages iii–1056–9 vol.3, May 2004.

[14] A. Shaikh, R. Tewari, and M. Agrawal. On the effectiveness of DNS-based server selection. In *IEEE INFOCOM*, Anchorage, AK, Apr. 2001.

[15] Team Cymru. The IP to ASN Mapping Project. (web page). http://www.team-cymru.org/Services/ip-to-asn.html.

[16] G. Varenni, L. Degioanni, F. Risso, and J. Bruno. WinPcap: The Windows Packet Capture Library. (web page). http://www.winpcap.org/.

[17] X. Xiao and L. M. Ni. Internet QoS: a big picture. *IEEE Network*, 13(2):8–18, 1999.

[18] S. Yang and G. de Veciana. Size-based adaptive bandwidth allocation: Optimizing the average QoS for elastic flows. In *Proceedings of IEEE INFOCOM '02*, New York, NY, June 2002.

[19] J. A. Zinky, D. E. Bakken, and R. E. Schantz. Architectural support for quality of service for CORBA objects. *Theory and Practice of Object Systems*, 3(1):55–73, April 1997.