OneClick: A Framework for Capturing Users' Network Experiences*

Cheng-Chun Tu†, Kuan-Ta Chen†, Yu-Chun Chang‡, and Chin-Laung Lei‡

†Institute of Information Science, Academia Sinica
‡Department of Electrical Engineering, National Taiwan University
{williamt,ktchen}@iis.sinica.edu.tw, congo@fractal.ee.ntu.edu.tw, lei@cc.ee.ntu.edu.tw

ABSTRACT

To learn about users' experiences in using network applications, we present a lightweight, non-intrusive, and efficient framework called `OneClick`. The framework only requires a subject to click a dedicated key whenever he/she feels dissatisfied with the quality of application in use. By aligning the collected traffic trace and the click process, we apply Poisson regression to analyze the relationship between click events and network QoS factors. The result shows that `OneClick` can be used to assess the effect of network factors on users' perceptions and compare the users' experiences with different types of design or applications.

Categories and Subject Descriptors:
H.1.2 [Models and Principles]: User/Machine Systems—*Human factors*; H.4.3 [Information Systems Applications]: Communications Applications—*Computer conferencing, teleconferencing, and videoconferencing*; H.5.1 [Information Interfaces And Presentation]: Multimedia Information Systems—*Evaluation/Methodology*

General Terms: Design, Experimentation, Human Factors

Keywords: Human Factors, Human Satisfaction, Network Experience, Poisson Regression, Quality of Service, VoIP

1. MOTIVATION

Providing high QoS for users of network applications is one of the most challenging tasks of network researchers because of limited resources. As the service requirements of network applications shift from high throughput to high media quality, interactivity, and responsiveness, *the definition of QoS is becoming more complicated and multidimensional*. For example, the QoS requirements of VoIP applications at least contain sound fidelity, voice loudness, noise level, echo level, and conversational delay; and the QoS requirements of online gaming at least contain interactivity, responsiveness, and consistency. While it is not difficult to measure any single dimension of the QoS levels, *how to capture users' perceptions when they are using network applications remains an open question*. For instance, suppose one network scenario provides a QoS setting of $(10, 15, 20)$, which represents the consistency level, interactivity level, and responsiveness level respectively, and another network provides a QoS setting of $(20, 15, 10)$. Which network configuration is "better" from the user's perspective remains to be investigated.

One common way to learn about users' network experiences is to ask them to complete a questionnaire after they use an application. However, this survey method cannot capture users' perceptions *in real time* as it is performed afterwards, and people are subject to the recency effect [1], i.e., the experience in the most recent experience dominates users' ratings. In addition, little information is captured because each subject only contributes one score in each test.

To overcome the above problems, we have developed a framework to capture users' perceptions of network applications. *We call the framework `OneClick`, because users are asked to click a dedicated button whenever they feel dissatisfied with the quality of the application in use.* `OneClick` is particularly effective because it is *lightweight, non-intrusive, and efficient*. It is lightweight because it does not need large-scale or expensive deployment of resources to perform experiments. It is non-intrusive because the user does not need to pay much attention to the scoring procedure, so his/her flow experience can be maintained. Finally, it is efficient because it captures many samples for a subject in each test. We detail the methodology in Section 2 and present a simple demonstration in Section 3.

2. METHODOLOGY

The `OneClick` framework comprises two phases. The first phase involves setting up experiments to collect users' feedback, and the second analyzes the collected raw data to summarize users' perceptions under different settings.

User experiments. The experiments can be performed on any computer equipped with a traffic monitor and a key logger. We ask a subject who is using a real-time interactive application, e.g., conferencing or gaming, to click a dedicated button whenever he/she feels dissatisfied with the quality of the application in use. Here, the quality of the application refers to any QoS dimension that might make the subject unhappy. For example, it could be poor voice quality or conversation smoothness in conferencing, or screen freezing problem, status inconsistency, or slow responsiveness in online gaming. Users do not need to be well-trained to participate in the experiments as only an intuitive click action is required.

Data analysis. We apply Poisson regression [2] to model the relationship between network factors and the click rate by treating the former as the predictors and the latter as a dependent variable. The click rate is the average number of times the subject clicks the button in one second. Assume the click rate is $C(t)$ and the network factors are $N_1(t), N_2(t), \ldots, N_k(t)$ at time t. Then, the Poisson regression equation is

$$\log(C(t)) = \alpha_0 + \alpha_1 N_1(t) + \ldots + \alpha_k N_k(t), \qquad (1)$$

*This work was supported in part by Taiwan Information Security Center (TWISC), National Science Council of the Republic of China under the grants NSC 96-2219-E-001-001, NSC 96-2219-E-011-008, and NSC 96-2628-E-001-027-MY3.

SIGCOMM'08, August 17–22, 2008, Seattle, Washington, USA.
ACM 978-1-60558-175-0/08/08.

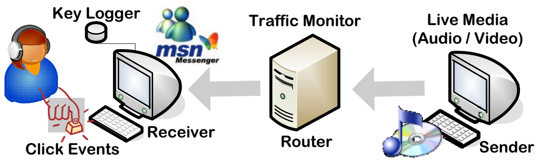

Figure 1: Network topology of the experiment.

where α_i denotes the Poisson regression coefficients, which are estimated by the maximum likelihood method.

Note that users may not always respond quickly, i.e., they may unintentionally delay the click actions after they become aware of the degraded application quality. To compensate for this effect, we shift the click event process. We search for the average delay time d_{avg} by fitting the regression model for network factors and click event processes with different time lags, where d_{avg} is computed as

$$argmin_d\{\text{deviance of (1) by replacing C(t) with C(t+d)}\}.$$

After obtaining the best model fit, we compute the expected click rate as $exp(\beta_0 + \beta_1 N_1 + \ldots + \beta_k N_k)$, where β_i is the Poisson regression coefficients by fitting $C(t + d_{avg})$ with $N_i, 1 \leq i \leq k$.

3. PRELIMINARY RESULTS

To illustrate the effectiveness of the `OneClick` framework, we apply it to compare users' network experiences in using AIM and MSN Messenger for *live* audio transmissions. The network setup is shown in Fig. 1. The router is configured to automatically change the network settings every 10 seconds. For each reconfiguration, it randomly sets the network delay within $(0, 2000)$ ms and the loss rate within $(0\%, 20\%)$. A song is played by the sender host, transmitted to the receiver via AIM or MSN Messenger, and played on the subject's earphone attached to the receiver host. We asked three computer science students to perform the experiments, where each experiment lasts for 6 minutes. Subjects are asked to press the space key whenever he/she feels unhappy about the music quality, without knowing the current network settings.

We incorporate three network factors, namely, the network delay, the delay jitter (i.e., the standard deviation of delays), and the loss rate, into the Poisson regression. The time lag that yields the best model fit is 1.8 seconds. Figure 2 shows the expected click rates computed based on different delay jitters and loss rates, as these two factors are more significant than the remaining delay factor. From the figure, we observe that AIM leads to a lower click rate than MSN Messenger when the loss rate is less than 20%. However, when the loss rate is 30%, AIM leads to a higher expected click rate than MSN Messenger. This indicates that AIM is less tolerant of an extremely high packet loss rate than MSN Messenger, even though it is more tolerant at low packet loss rate.

We define a "comfort region" as the set of network scenarios that leads to an expected click rate lower than a certain threshold. From Fig. 3, we can see the comfort regions of the two applications are different regardless of the click rate threshold. When the click rate threshold is 0.4 or 0.6, the smaller comfort regions of MSN Messenger indicates that its quality is not as good as that of AIM. However, if we set the click rate threshold at 0.8 or higher, MSN Messenger and AIM are comparable. MSN Messenger is more tolerant of packet loss and AIM is more tolerant of network delay jitters. This may be because the applications adopt different codecs. We leave the explanation of this behavior to a future study.

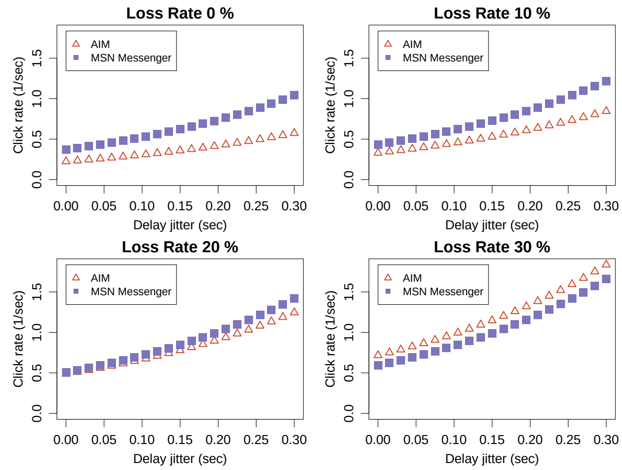

Figure 2: Expected click rates computed based on different delay jitters and loss rates.

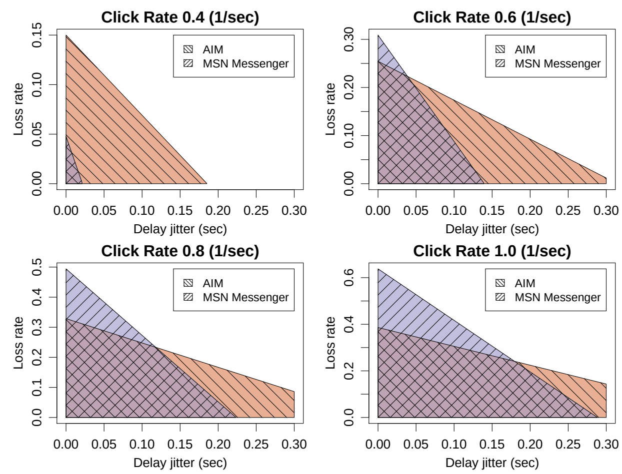

Figure 3: Comfort regions computed based on different delay jitters and loss rates.

4. CONCLUSION

In this poster, we present a framework called `OneClick`, a framework for capturing users' network experiences. The framework only requires a subject to click a dedicated key whenever he/she feels dissatisfied with the quality of application in use. By aligning the collected traffic trace and the click process, we apply Poisson regression to analyze the relationship between click events and network QoS factors. We have shown that `OneClick` can be used to assess the effect of network factors on users' perceptions and compare the users' experiences with different types of design or applications. We plan to incorporate dynamic delayed responses and user bias into the analysis procedures to improve the reliability and accuracy of our proposed framework.

5. REFERENCES

[1] A. Watson and M.A. Sasse. Measuring perceived quality of speech and video in multimedia conferencing applications. *Proceedings of the sixth ACM international conference on Multimedia*, pages 55–60, 1998.

[2] F.E. Harrell. *Regression Modeling Strategies: With Applications to Linear Models, Logistic Regression, and Survival Analysis.* Springer, 2001.