Characterizing Human Effort in Wireless Voice Over IP

Jahan A. Hassan[1] Mahbub Hassan[1] Sajal K. Das[2]

[1] University of New South Wales, Australia
{jahan,mahbub}@cse.unsw.edu.au
[2] University of Texas at Arlington
das@cse.uta.edu

Technical Report
UNSW-CSE-TR-0916
September 2009

THE UNIVERSITY OF
NEW SOUTH WALES

School of Computer Science and Engineering
The University of New South Wales
Sydney 2052, Australia

Abstract

Skype Voice Over IP (VoIP) traces from an experimental WiFi network were analyzed to detect and characterize user efforts that go into these calls. Our analysis shows that users have a very low tolerance threshold when it comes to putting efforts for getting the conversation going (they prefer rather effort-less conversation). A wireless VoIP session is highly likely to be abandoned prematurely by the user if the effort threshold is exceeded during the call. Our results also suggest that after exceeding the effort threshold, users are likely to spend quite a bit of time in the call before finally abandoning it. These effort patterns are found to be consistent across multiple users, with the actual value of the effort threshold being sensitive to the user. An important outcome is that it is possible to reliably generate warnings for calls that are going to face premature ending by simply monitoring the number of times the user has put efforts into the call. Besides reliability, these warnings can be generated well in advance giving plenty of time to network controllers for possible repair of the wireless connection and avoidance of premature call ending. Using the effort data captured from our experimental network, we conduct discrete event simulations of a WiFi VoIP network to evaluate the effectiveness of dynamic resource allocation in addressing such warnings. The experiments show that resource allocation schemes which are capable of exploiting the long warning lead times of effort-based predictions, can find additional resources with a high probability.

1 Introduction

Although wireless VoIP is set to become a major telephony market, it will probably continue to face the prospect of occasional link quality issues that are not existent in traditional telephone (PSTN) networks. If there are link quality issues, for example a link experiencing interference from a nearby WiFi access point operating in the same frequency, users naturally put some human-level effort to continue the conversation. These efforts are basically user attempts to retransmit lost speech at the human language level using words or phrases such as "I can't hear you, can you repeat it please"?

Technically, it may be possible for humans to continue conversation for a long time over a bad quality link, but it will require a lot of efforts on their parts. Although capable, the users may not be willing to put such efforts. Understanding how users of wireless VoIP spend efforts during a good and a bad quality VoIP link may reveal insights that could be useful in detecting the onset of user irritation in an ongoing call. More importantly, such insights may be useful in designing tools and techniques that can address the perennial quality issue associated with wireless VoIP.

Despite there being many studies to understand how users perceive the quality of a VoIP session experiencing network quality problems (e.g., packet loss, delay, jitter etc.), work on understanding user effort to deal with a bad quality link is rare. The intent of our study, therefore, is to conduct a systematic experiment with real users and investigate their effort patterns for both good and bad quality VoIP calls. Specifically, we set up an experimental WiFi Skype VoIP network and employ seven students to complete 27 VoIP calls spread over a period of 3 months. The entire conversations are recorded and analyzed manually at the human language level to identify the occurance of each and every effort instance in every conversation.

Our effort analysis reveals quite interesting insights into user behaviour when it comes to putting efforts in a VoIP session. Our effort data shows that users have a very low tolerance threshold when it comes to putting efforts for getting the conversation going. In other words, users basically prefer rather effort-less conversation. A wireless VoIP session is highly likely to be abandoned prematurely by the user if a given effort threshold is exceeded during the call. Our results also suggest that after exceeding the effort threshold, users are likely to spend quite a bit of time in the call before finally abandoning it. These effort patterns are found to be consistent across multiple users, with the actual value of the effort threshold being sensitive to the user.

An important outcome is that it is possible to reliably generate warnings for calls that are going to face premature ending by simply monitoring the number of times the user has put efforts into the call. Besides reliability, these warnings can be generated well in advance giving plenty of time to network controllers for possible repair of the wireless connection and avoidance of premature call ending. Using the effort data captured from our experimental network, we conduct discrete event simulations of a WiFi VoIP network to evaluate the effectiveness of dynamic resource allocation in addressing such warnings. The experiments show that resource allocation schemes which are capable of exploiting the long warning lead times of effort-based predictions, can find additional resources with a high probability.

The rest of the paper is organised as follows. Related work is reviewed in Section 2. We explain the experimental setup and data collection process in Section 3. The effort analysis and the predictability of premature call completion are presented in Section 4. Simulation-based performance evaluation of the prediction of premature call ending, and the implications of allocating extra resources to those predicted calls are presented in Section 5. Finally, we draw our conclusion in Section 6 followed by a discussion of possible future work.

2 Related Work

There is a significant body of prior work in the literature to assess the quality of VoIP calls, either objectively [5,9] or subjectively as perceived by the user [8]. However, most of these works are concerned with assessing the quality (or user perception of) a call that has already completed, and hence cannot capture the real-time user behaviour as a call progresses. The notable exceptions are the works by Chen et. al. [3,4]. In their Oneclick work [4], the authors propose a method to capture real-time user perceptions by requiring the user to press a button whenever the user feels unhappy about the quality of the call. In their other work [3], the authors propose a user satisfaction index which can be calculated on-line by monitoring the packet traces as the call progresses. Our work has a similar spirit to that in [3] in the sense that we also attempt to gauge user reactions in real-time during a call, but we differ in our approach and application. Our approach is unique in trying to identify and measure the efforts put in by the users in a VoIP call. The application is also unique in exploiting user tolerance to efforts to avoid the undesirable event where a user quits a VoIP session prematurely due to dissatisfaction.

3 Data Collection

All call and user effort data were collected from an experimental WiFi VoIP testbed at the University of New South Wales. The testbed set-up is shown in Figure 3.1. An WiFi access point (AP) was built using a Dell Latitude D800 laptop running Ubuntu Linux OS and a D-Link wireless 108G multiple-input, multiple-output (MIMO) card attached to it for wireless ac-

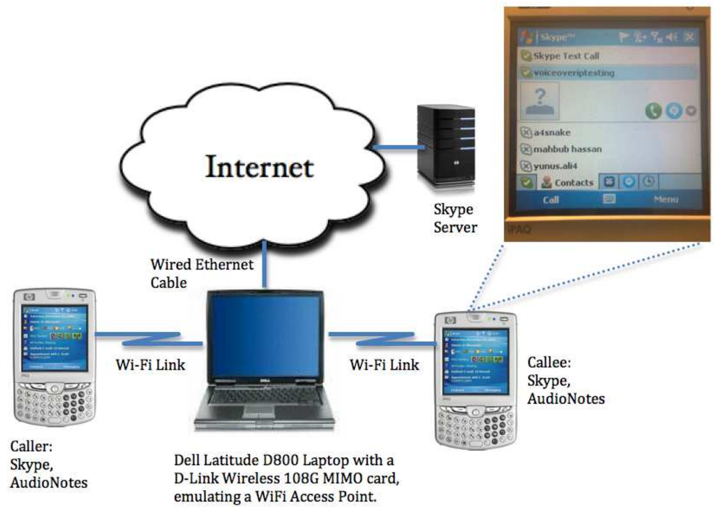


Figure 3.1: WiFi VoIP testbed.

cess. A DHCP server (udhcpd) was installed so that the wireless stations (WiFi handsets) connecting to the WiFi network can obtain IP addresses for Internet communication. Using a wired Ethernet interface, the AP was connected to the Internet via the university's network, so users could access the Skype server on the Internet for establishing the VoIP calls. HP iPAQ hw6965 Windows Mobile Pocket PCs [7] were used to make the wireless VoIP calls. These iPAQs have WiFi interface and runs the Skype client [10] freely available for Windows Mobile handsets. We installed AudioNotes [12] in the iPAQs to record the outgoing audio traces of every VoIP call (audio was recorded in .WAV format).

Seven students took part in a total of 29 Skype VoIP calls. At any given time, two students used the iPAQs in two different rooms (201B and 217) of the same floor (see Figure 3.2 for the floor plan) to establish and complete a VoIP call. The Dell Laptop with the D-Link card is located in room 217G. Around 10 meters away, the kitchen (Room 215) houses a microwave which is used frequently by students and staff. Besides our experimental AP, the floor also houses our department's WLAN APs, which are used by students and staff for their work.

To make the VoIP calls, the students first turned on their WiFi network connection to connect to our testbed WiFi network. Once connected, they log on to previously set up Skype accounts in the Skype login server on the Internet. Once both of them are logged on to Skype, the Skype screen on both iPAQ shows each Skype user being on-line. At this point, one student called the other by using the other phone's Skype account name as shown on the callee's Skype screen (a screenshot of Skype is shown in Figure 3.1 as

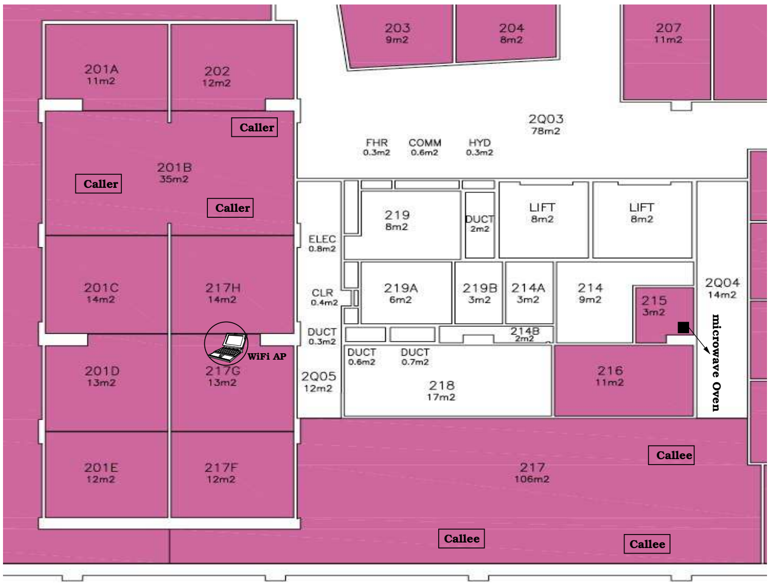


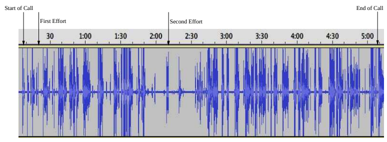
Figure 3.2: Floor plan of the testbed.

a blown-up screen). It is to be noted that for every Skype call our students made, the signal strength was found adequate to successfully establish skype calls between the pairs (despite the fact that for many calls the quality of the call was not good enough to continue the call as normal). For each call, students were asked to discuss a research paper for about five minutes or so. They were given the liberty to hang up prematurely if they thought it was getting difficult for them to continue conversation. Before starting the Skype call, the students started call recording on their iPAQs using AudioNotes to ensure we can capture the entire call and do not miss portions of it due to recording startup delay. This is why we have some leading audio in each recording that do not belong to the call. We have discarded these portions in our data analysis.

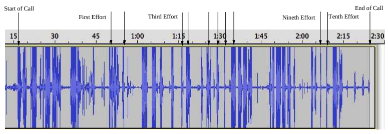
At the end of each call, whether completed naturally or prematurely, each student gave a rating (opinion score) of the call between 1 and 5 where 5 is Excellent. Therefore, for each VoIP session, we collected two audio traces and two ratings, one from each of the two participating students.

Definition 1 *User Effort, or simply Effort, is defined as the speakers attempt to recover lost speech using keywords or phrases like “sorry?”, “hello?”, “Can you repeat it?” and so on, with respect to the context in which those words were spoken.*

Data on user effort were collected manually by listening and scrutinizing the replay of each of the 58 audio files. The audio files, which were saved in .WAV format, were played on the computer using QuickTime Player.



(a) A naturally ending call.



(b) A prematurely ending call.

Figure 3.3: Detection of user efforts in wireless VoIP calls (arrows show the timing of the efforts as they occurred in the speech)

For each audio trace, the listener recorded the time of each effort. Use of a human in this way ensured that our effort detection was reliable and accurate. We contemplated automating the process, but decided against it because we have found that existing audio mining tools are not very reliable and they are unlikely to understand the context in which certain keywords are used. For example, the word "hello?" could be used as part of natural conversation as well as when the user is attempting to recover lost speech. A human listener can reliably distinguish between the two. A summary of all such keywords and phrases that were used by the users as an effort to recover speech are given in Table 3.1.

Figure 3.3 shows the occurrences (timing) of efforts for two particular audio traces, one for a naturally ending and one for a prematurely hung-up call (the audio signals were captured using Audacity audio tool [1]). Calls tagged with effort timing are used to guide the discrete event simulation experiments described later in the paper (Section 5). The listener also identified the call ending mode of each call as either naturally ended or completed prematurely by scrutinizing the end words (or phrases) of the calls. Once the effort timings are extracted for a call, we derive the effort count as the total number of efforts found in a call. These effort counts are analyzed in the following section.

4 Effort Analysis

In this section, we analyze user behaviour in naturally and prematurely ending calls with respect to the effort they (the users) put into the calls. A key objective is to establish whether it is possible to predict the ending mode of a call by analyzing the effort pattern found in the speech.

4.1 Effort Distributions

Analysis of the recorded speech traces showed that out of 58 user sessions, only 27 ended 'naturally'. For the remaining 31 sessions, users 'prematurely' quit the service out of frustration due to poor quality. It is interesting (and

Table 3.1: Summary of keywords and phrases indicating human efforts

“Sorry?”, “Did you say anything back then?”, “Can you repeat it?”, “hello?”, “Can you hear me?”, “What’s that?”
“What did you say?”, “I can’t understand”, “What did you ask?”, “I can’t hear you”, “Nothing coming out!”, “hi?”, “hang up?”
“Could not hear anything”, “Sorry, what was that?”, “It cuts in and out”
“I can’t hear what you’re saying”, “What’s that?”, “What?”
“Too bad, hang up”, “That’s it!”, “Could not hear you”
“Can you repeat? it dropped out”, “This is really bad” , “Words are lost”
“could not hear the full sentence”, “may be we finish, can’t hear”
“could not hear you, dropped out”, “hang up?”, “Got cut off”
“huh?”, “What?”, “Unclear”, “Sorry, missed that”
“sorry, say again?”, “Did not get that”, “You are breaking out”
“Can’t quite hear you”, “A bit unclear”, “I’m losing half the sentence”
“Can’t hear you at all”, “Sorry what are you saying?”, “I did not hear you”
“sorry missed that, say again?”, “you are dropping out”, “You there?”
“sorry? Did not get you”, “It’s getting worse”, “I’m not getting you”
“Hello! I’m here!”

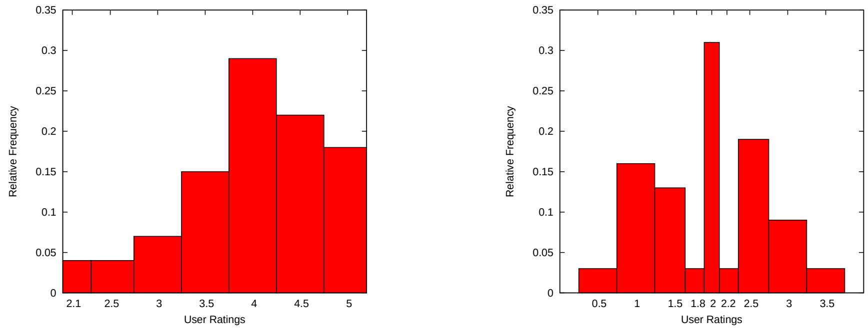
alarming) to see that such a high percentage of calls (53%) are abandoned prematurely by the users. Figure 4.1 shows the probability distributions of user rating of the calls. As expected, users poorly rated the calls they ended prematurely and expressed improved satisfaction for the calls they completed naturally. This confirms that users are likely to abandon a call if they are not satisfied with the quality of the call.

To gain a deeper understanding of the relationship between the mode of call completion and the user effort, we plot the probability distributions of the effort count, i.e., the number of times the user had to ‘put effort’, of both naturally and prematurely ended calls in Figure 4.2. We discover the following:

1. For naturally ended calls, effort count distribution has a very short tail (the maximum effort count observed was only four). In other words, users basically want effort-less conversation.

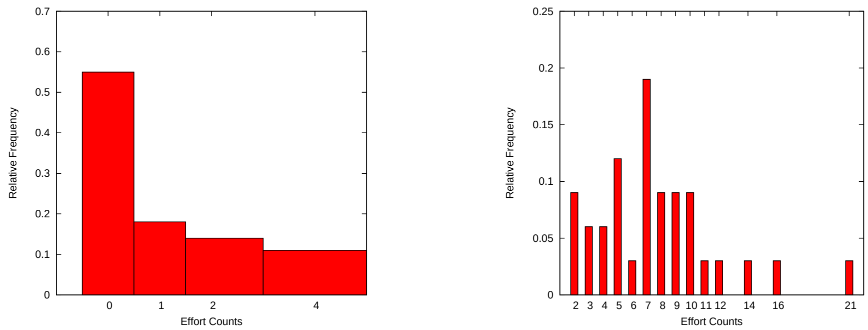
2. Opposite to naturally ending calls, prematurely ending calls exhibit a very long tail for their effort count distribution (effort count was as high as 21 in some prematurely ended calls).

These discoveries suggest that (i) on-line effort measurement can reliably predict whether the user would end the call prematurely, and (ii) the prediction about premature ending can be made quite early in the call leaving plenty of warning time for network control functions to improve the link quality in an attempt to avoid premature call ending in the system. We will



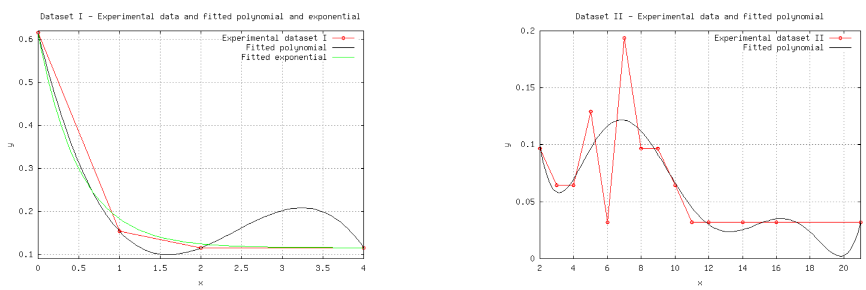
(a) Naturally Ending Calls (b) Prematurely Ending Calls

Figure 4.1: Probability distribution of user ratings (opinion scores).



(a) Naturally Ending Calls (b) Prematurely Ending Calls

Figure 4.2: Probability distribution of effort counts.



(a) Naturally ending calls (b) Prematurely Ending calls

Figure 4.3: Curve Fitting for Effort Counts.

provide a more quantitative analysis of the accuracy of such effort-based premature call ending predictions and the warning time that can be achieved later in the paper.

Given the experimental results depicted in Figures 4.2 which were obtained for frequencies (Y axis) of different effort count values (X axis), it is worthwhile to analyze these two cases: calls that end naturally and those that terminated prematurely. In the case of naturally ended calls (NEC), the frequency values can be best approximated by an exponential function of the total effort (TE) as follows (also depicted in Figure 4.3(a):

$$NEC(TE) = 0.5 \times e^{-2 \times TE} + 0.11538 \tag{4.1}$$

where the constant term 0.11538 is due to the limit that is to be achieved for TE = 4, that is NEC(4) = 0.11538, while the coefficient of the exponential is given by the value of the data set for TE = 0, that is NEC(0) = 0.61538, from which the constant term is subtracted. Finally, a coefficient -2 for the exponent has the effect of lowering the curve to fit the frequency values in the central area of the domain TE.

In the case of prematurely ended calls (PEC), a solution based on an exponential function cannot provide a satisfying result. Thus, polynomials of degrees between 1 and 10 were fitted to the curve using GNU Octave. The best fit is yielded by the following polynomial (also depicted in Figure 4.3(b)):

$$\begin{aligned} PEC(TE) \quad = \quad & 9.1513e^{-07}TE^6 - 6.5129e^{-05}TE^5 + \\ & 1.8035e^{-03}TE^4 - 2.4357e^{-02}TE^3 + \\ & 1.6394e^{-01}TE^2 - 5.0242e^{-01}TE + \\ & 6.1676e^{-01} \end{aligned} \tag{4.2}$$

Although the polynomial in Equation (4.2) provides an acceptable approximation for values in the intervals [2, 4] and [9, 16], the spikes in the interval [5, 8] are not represented by this solution.

4.2 Effort-based Prediction of Premature Calls

Next we turn to analyze the warning time that can be achieved with effort-based prediction of premature call ending. For effort threshold of 2 (global static threshold), we find that the average alarm-to-hangup time is 93 seconds, meaning that on average, providers would get more than one and half a minute to improve the network performance before the user abandons the call. The warning time distribution for all correctly predicted prematurely ending calls are shown in Figure 4.4. We observe that for 70% of the times, the prediction algorithm gave warnings of 60 seconds or more and the most dominant warning period is 100-200s. We also find that warnings are unlikely to be too close to the hangup time, e.g., less than 15s has a probability

Table 4.1: Impact of effort threshold on prediction accuracy.

Effort Threshold	Accuracy	False Alarm
0	100%	40%
1	100%	25%
2	90%	11%
3	84%	11%
4	78%	0%
5	65%	0%

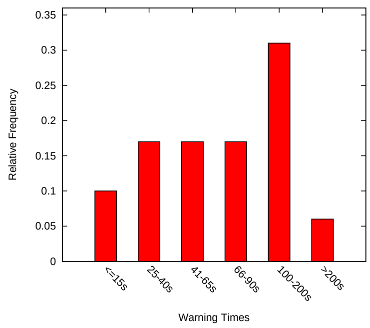


Figure 4.4: Probability distribution of warning times.

less than 10%. The promise of such advance warning is encouraging because the longer the user hangs on to the call, the more time the network gets to secure additional unused resource for these calls, and the higher the chances of avoiding the premature ending of the call. The quantitative performance of effort-threshold based warning in avoiding premature ending of a call is captured by discrete event simulation in the following section.

The effort analysis in the previous section revealed that users want rather effort-less conversation. This revelation can be exploited to predict whether a call is likely to face a premature ending by implementing a simple effort

```
Input Effort_Threshold;
For each call
  set Total_Effort = 0;
  Do While(Total_Effort<=Effort_Threshold)
      wait until effort detected
         Total_Effort++;
         If (Total_Effort>Effort_Threshold)
              Generate Warning for Premature ending
              exit
```

Figure 4.5: Premature Call Ending Prediction Algorithm

Table 4.2: Effort Threshold for Individual Users.

User	Effort Threshold	Comment
User1	6	Minimum effort count was 7 for prematurely ending calls. All calls by this user were Prematurely ending.
User2	1	Effort count was 2 for this user's prematurely ending calls, however, a small fraction of naturally ending calls had an effort count over 2 thus these calls will get picked for warning wrongly.
User3	2	There were overlapping effort count of 2 between naturally and prematurely ending calls. However, majority of the prematurely ending calls had an effort count over 2, thus they can be correctly picked for warning.
User4	1	All the prematurely ending calls had an effort count over 1. A small portion of naturally ending calls had an effort count over 1 thus these calls will get picked for warning wrongly.
User5	3	The lower bound of effort count in prematurely ending calls was 4, and the effort counts in naturally ending calls were all below 3. Thus the choice of 3 as the effort threshold was obvious.
User6	4	The lower bound of effort count in prematurely ending calls was 5, and the effort counts in naturally ending calls were all below 4. Thus the choice of 4 as the effort threshold was obvious.
User7	7	The lower bound of effort count in prematurely ending calls was 8, and the effort counts in naturally ending calls were all below 7. Thus the choice of 7 as the effort threshold was obvious.

threshold based algorithm which works as follows. When a call is admitted, the predictor starts to count the number of efforts encountered so far in the call and compared it with a effort threshold. The call is predicted to be abandoned prematurely, if the effort count exceeds the threshold. This is shown in Figure4.5.

We apply the above mentioned algorithm to our 58 call traces which were later augmented with the timings of efforts in the speech. The prediction time is recorded so we can obtain the warning time (time elapsed from the moment the warning was generated until it was finally hung up by the user). We compute the prediction accuracy as the fraction of prematurely hung up calls that were predicted accurately. *False alarm* on the other hand is computed as the fraction of naturally ended calls that were falsely predicted to be abandoned prematurely. Table 4.1 shows the prediction accuracy achieved along with the false alarm rates for 6 increasing effort thresholds. As expected, both accuracy and false alarms start to drop as we increase effort threshold. Thresholds 0 and 4 give us two extreme results, as 100% accuracy is achieved at zero threshold, while false alarms are completely eliminated by setting the threshold to 4. However, neither extremes are good. To achieve 100% accuracy, the minimum false alarm rate that can be achieved is 25% (for a threshold of 1). On the other hand, to achieve zero false alarm in the system, the maximum accuracy that can be achieved is 78%. Perhaps, a more balanced control would be to achieve a high accuracy with a reasonable false alarm rates (false alarms would cause unnecessary resource allocation to otherwise 'healthy' calls resulting in scarcity of resources). For example, setting the effort threshold to 2 would yield an accuracy of 90% with a false alarm rate near 10%.

To investigate whether the prediction performance can be improved further by customizing the threshold for different users, we studied the probability distribution of effort counts for each user individually. We found that although the 'tail length disparity' (i.e., the fact that probability distribution of naturally ending calls have a much shorter tail than that for prematurely ending calls) is consistent across all users (Figures 4.6 to 4.12 show the distributions of two different users), different users do exhibit different effort thresholds. For example, for the user of Figure 4.10, it appears that a threshold of 3 would be more appropriate as the minimum effort spent by this user is 4 for any prematurely abandoned call. However, for the user of Figure 4.12, a threshold of 7 is more appropriate. Such significant threshold disparity among different users suggest that prediction algorithms that are capable of adaptively switching to different thresholds for different users are likely perform better than those which use a fixed global threshold for all users. By applying such an adaptive threshold algorithm which uses different optimum threshold for different users, we were able to increase the prediction accuracy from 90% to 97% while reducing the false alarm rate from 11% to 7%. The customised threshold for different users are shown in Table 4.2 with explanations on the choice of the thresholds.

5 Simulation

Effort-based prediction that a call is going to end prematurely due to bad call quality can be of use only if something can be done to improve the quality of the call, with the hope that the call will not end prematurely. In many cases, injecting extra resource can solve the quality problem. We have carried out extensive simulation studies to explore how the interactions of resource availability and the quantity of extra resource needed for improving the call

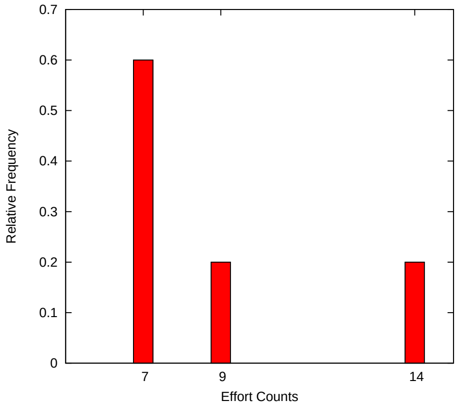
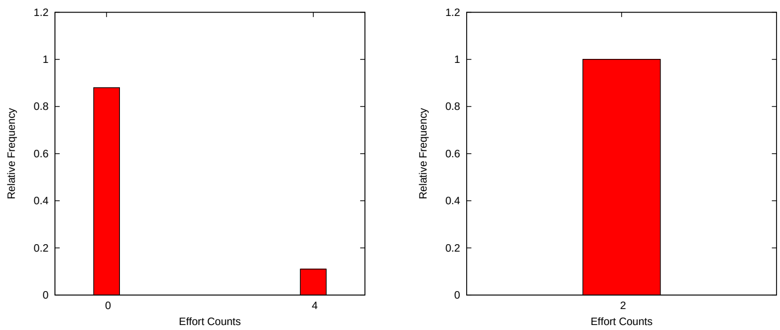


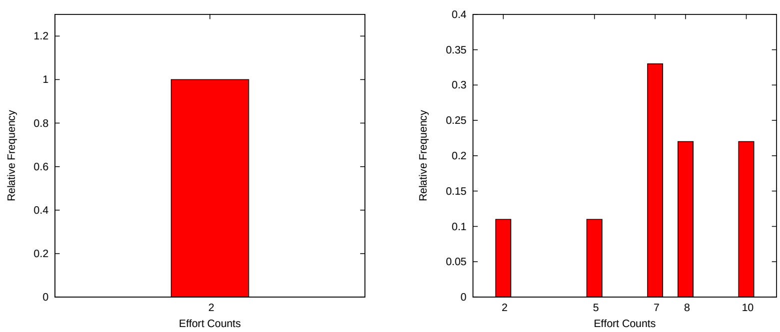
Figure 4.6: Effort count distribution for User1 (Prematurely Ending Calls. This user had no call ending naturally)



(a) Naturally Ending Calls

(b) Prematurely Ending Calls

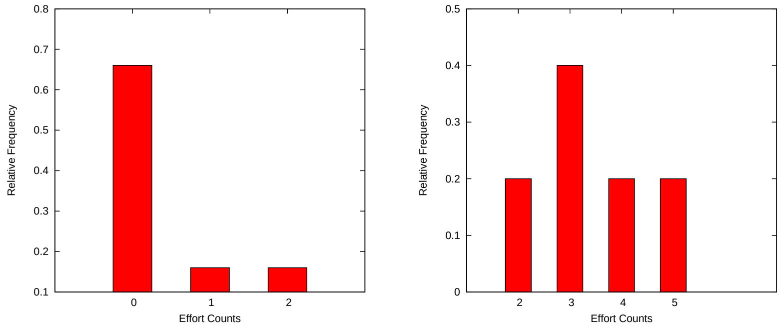
Figure 4.7: Effort count distribution for User2.



(a) Naturally Ending Calls

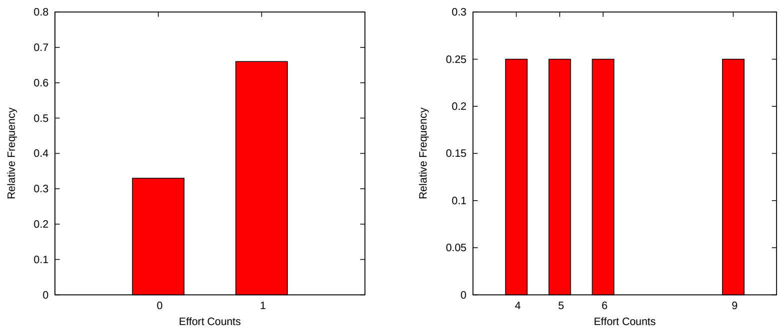
(b) Prematurely Ending Calls

Figure 4.8: Effort count distribution for User3.



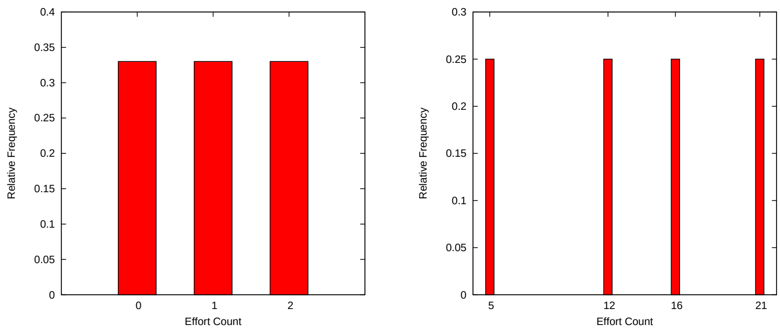
(a) Naturally Ending Calls (b) Prematurely Ending Calls

Figure 4.9: Effort count distribution for User4.



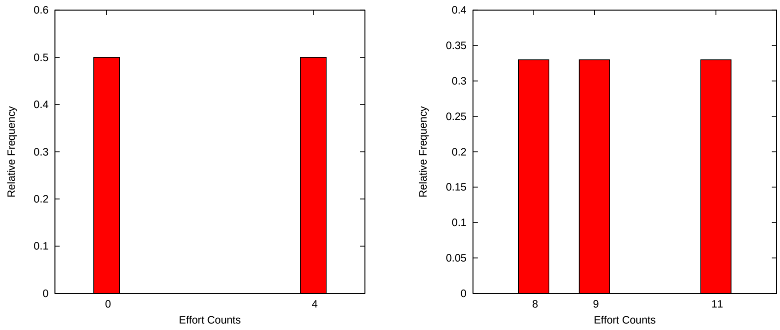
(a) Naturally Ending Calls (b) Prematurely Ending Calls

Figure 4.10: Effort count distribution for User5.



(a) Naturally Ending Calls (b) Prematurely Ending Calls

Figure 4.11: Effort count distribution for User6.



(a) Naturally Ending Calls (b) Prematurely Ending Calls

Figure 4.12: Effort count distribution for User7.

quality impact the overall performance improvement of a WLAN in terms of converting warning calls (calls that are predicted for premature ending) into naturally ending calls in light of the help of our effort-based prediction. We also study what implications this allocation of extra resources have on the incoming calls, i.e., call blocking due to lack of adequate resources as on-going warning calls get extra resource allocation. These interactions are very important for the providers as these will influence their decisions of when and whether or not to inject extra resources to the warning calls.

We have written a discrete event simulator in C to simulate a wireless LAN where users share the limited bandwidth managed by an admission controller to ensure some minimum call quality. In particular, the admission controller does not accept more than 17 calls following the recommendation in [11] which found that accepting more calls would increase the delay beyond 213ms (for an inter-poll period of 60ms). Calls arrive to the LAN following a Poisson process with an arrival rate of λ calls per second. The properties of the calls in terms of their length, pattern of user effort within the call, and whether the call is supposed to end prematurely or not, are guided by the results obtained from our experimental study. More specifically, when it is time for a call to arrive, a call is randomly selected from the pool of 58 calls collected from our experimental WiFi testbed (see Section 3).

Warnings are generated based on an effort threshold of 2, i.e., if a third effort is encountered in a call, the resource manager receives a warning. Upon receiving a warning, the resource manager allocates an additional amount (Δ) of bandwidth to the distressed call if there is enough bandwidth available in the system. For the event that no resource is available when a warning is received, we implemented two different schemes, queueing and no queueing. In the queueing scheme, the network stores the warning (request for resource) in a first-in-first-out (FIFO) queue and serves the warnings from the queue when some resources are freed up (when some calls end). In

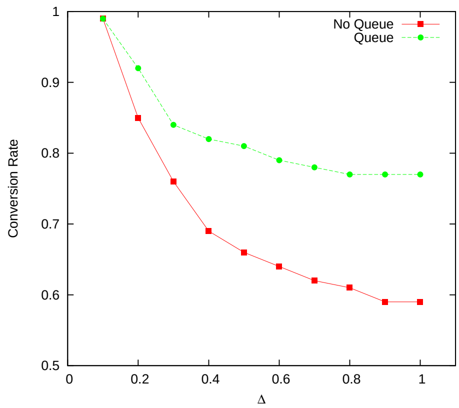


Figure 5.1: Probability of resource availability.

the no-queueing scheme, the system simply ignores a warning if there is no resource available to allocate to the call at that time. The queueing scheme is more complicated, but it can really take advantage of the long warning periods that are possible with effort-based predictions.

The amount of additional resource that is allocated to a distressed call upon receiving a warning is denoted by Δ. In actual systems, the exact value of Δ would depend on actual physical techniques used to repair the quality problem experienced by a given wireless link. For example, the bandwidth needed to support a forward error correction (FEC) [2] may be significantly less than if an additional channel is allocated for transmitting packets redundantly (e.g., the ones proposed for vehicular communications to boost the broadcast packet reception rates during periods of high levels of packets collisions [6, 13]). Δ is a key parameter that will affect the success rate of avoiding premature call ending. In our simulation we consider a range of values for Δ normalized to the bandwidth allocated to each call during call establishment.

As a function of Δ, Figure 5.1 shows the probability of resource availability when a warning is received from a distressed call (for an λ of 1 call arrival every 3 minutes). As expected, the system can respond more effectively to the warnings when less amount of resource is needed to address the quality problem. The more important observation here is that the probability of resource availability is significantly increased when the large warning period is captured via queueing. As a direct consequence, we see that queueing can achieve a much lower call quiting probability (CQP) in the system (see Figure 5.2), especially when Δ is large. It should be mentioned that the significant reduction in CQP is achieved with only a *minimal* increase in call blocking probability (see Figure 5.3).

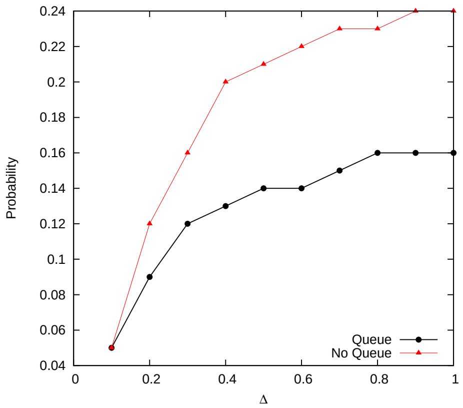


Figure 5.2: Call quitting probability.

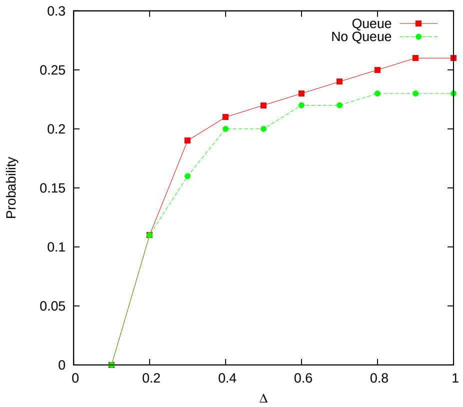


Figure 5.3: Call blocking probability.

6 Conclusion and Future Work

Using real users, we have analyzed and characterized user effort in wireless VoIP. We have discovered a consistent effort pattern across multiple users. We have found that users have a very low tolerance threshold for putting efforts in VoIP calls. users are likely to abandon a wireless VoIP session prematurely if the effort threshold is exceeded during the call. In contrast, after exceeding the threshold, users are likely to spend a rather long time in the call before finally abandoning it. We have shown that these effort patterns can be exploited to generate reliable warnings well in advance to the user quitting the call. Using discrete event simulations, we have demonstrated that with appropriate dynamic resource management, the reliability and advance generation features of such effort-based warnings can significantly reduce call quitting probability in a wireless VoIP network.

There are several new directions for future work. In our current work, we have manually identified human efforts (primarily for reliability reasons). For practical systems, we need monitoring tools and techniques that can automatically identify all instances of human efforts in a VoIP call accurately and in real-time without any manual intervention, a task that is both interesting and challenging. Establishing how much resource allocation can reduce how much human effort is another interesting direction to pursue.

Acknowledgements

We would like to thank all the student participants who gave us their valuable time to take part in VoIP speech sample data collection. We also like to thank Giacomo Ghidini of Univ. of Texas at Arlington for his help with Matlab.

This research is funded by the Australian Research Council Discovery Project DP0881553.

Bibliography

[1] Audacity Cross-Platform Sound Editor, URL: http://audacity.sourceforge.net/. accessed on 30 July 2009.

[2] E. Altman, C. Barakat, and V. M. Ramos R. Queueing analysis of simple FEC schemes for IP telephony. In *Proceedings of IEEE INFOCOM*, pages 796–804, 2001.

[3] Kuan-Ta Chen, Chun-Ying Huang, Polly Huang, and Chin-Laung Lei. Quantifying skype user satisfaction. In *Proceedings of ACM SIGCOMM 2006*, Pisa Itlay, Sep 2006.

[4] Kuan-Ta Chen, Cheng Chun Tu, and Wei-Cheng Xiao. Oneclick: A framework for measuring network quality of experience. In *Proceedings of IEEE INFOCOM*, 2009.

[5] ITU-T Reccomandation P.107, The E-model, a computational model for use in transmission planning. http://www.itu.int/.

[6] F. Farnoud and S. Valaee. Repetition-based broadcast in vehicular ad hoc networks in Rician channel with capture. In *Proceedings of IEEE INFOCOM MOVE Workshop*, pages 1–6, 2008.

[7] HP iPAQ Pocket PC and Calculators, URL: http://welcome.hp.com/country/us/en/prodserv/handheld.html?jumpid=reg_R1002_USEN . accessed on 1 June 2009.

[8] ITU-T Reccomandation P.800, Methods for subjective determination of transmission quality, International Telecommunication union, 1996.

[9] ITU-T Reccomandation P.862, Perceptual evaluation of speech quality (PESQ), an objective method for end-to-end speech quality assessment for narrow-band telephone networks and speech codecs, Feb 2001.

[10] Skype Limited, URL: http://www.skype.com/, accessed on 7 Aug 2008.

[11] M. Veeraraghavan, N. Cocker, and T. Moors. Support of voice services in ieee 802.11 wireless lans. In *Proceedings of IEEE INFOCOM*, 2001.

[12] Audio Notes, VoIP recorder, Vito Technology. URL: http://vitotechnology.com/vito-audionotes.html, accessed on 1 May 2009.

[13] Q. Xu, T. Mak, and R. Sengupta. Vehicle-to-vehicle safety messaging in dsrc. In *Proceedings of the 1st ACM international workshop on Vehicular ad hoc networks*, pages 19–28, 2004.