Readiness of 802.11 Infrastructure for Mobile Learning Communities

Klara Nahrstedt, Zheng Zeng, Long Vu, Anjali Sridhar
University of Illinois at Urbana-Champaign, Urbana, IL, USA
{klara, zzeng2, longvu2, sridhar3}@illinois.edu

Ying Huang
Google, Inc. USA
Ying.huang.uiuc@gmail.com

Abstract: Advances in wireless technologies such as 802.11 have allowed students to communicate ubiquitously on our campuses and made it possible for them to exchange information through mobile devices such as cell-phones, tablets and laptops. In this paper, we explore the premise of Google Android phones as educational tools for undergraduate students in our educational institution, and creation of effective mobile learning communities through the usage of cell-phones. We present one year investigation, where we want to answer one of the many questions: is the WiFi communication infrastructure at the educational institution ready for massive phone deployment in and out of classroom? Through combination of discussions with undergraduate students, deployment of phones and the ‘mi-clicker’ learning application, and measurements of WiFi infrastructure during phone usage in selected classrooms, we present the complexity of the issues when answering the question about mobile learning communities on a university campus.

1. Introduction

We are seeing a ubiquitous deployment of mobile 802.11 technologies such as WiFi-enabled cell-phones, tablets and laptops in our everyday lives. Especially, on our university campuses students use mobile 802.11 technologies for their personal usage and access to various resources such as information about local events, restaurants, access to class websites, and sharing of local experiences with each other.

In this paper we explore mobile learning community (MLC) at the University of Illinois at Urbana-Champaign (UIUC) campus, consisting of undergraduate students, WiFi-based infrastructure over which the students communicate, and our educational institution that provides education to students (as shown in Figure 1). This mobile learning community (MLC) already uses WiFi 802.11 infrastructure everywhere on the UIUC campus to access educational servers and services via laptops and tablets. In this paper, we want to explore the following question: if we imagine that classes of MLC undergraduate students would get a WiFi-enabled phone as an educational tool from the educational institution, would the campus WiFi infrastructure be ready for this step?

Note that the infrastructure question is only one of the many questions that need to be answered to make MLC effective on our campuses, including operational, educational and staff issues (Nahrstedt et al. 2010). We have conducted one year study (August 2009 – August 2010) to find answers to the above question.

Through multiple facets of investigation, including questionnaires to users, deployment of large scale phones and the ‘mi-clicker’ educational application in selected classes running over WiFi infrastructure, and WiFi/software measurements and evaluation, we will describe the complexity of the answers to the WiFi infrastructure question. Furthermore, we will present lessons learned towards the next steps of effective mobile learning. In Section 2, we describe the methodology to explore answers to the infrastructure question. In Section 3, we elaborate on the student model, the ‘mi-clicker’ educational application and the usage model that we had considered when measuring and evaluating the WiFi departmental infrastructure. Section 4 presents the
deployment efforts in the systems programming cs241 class and student evaluation of the deployment efforts in cs241. Section 5 presents the classroom WiFi experimental setup and infrastructure measurements during the deployment efforts in cs241 class. Section 6 discusses related work in this area. We conclude our answers in Section 7.

Figure 1: Mobile Learning Communities

2. Methodology of Investigation

To understand mobile learning communities and finding out how cell-phones hold as educational tools in our undergraduate classrooms, we have considered a mixture of approaches to find answers to the WiFi infrastructure challenge:

1. Development of Educational Application: We have developed an educational application, called mi-clicker, on the WiFi-enabled phones. This application was designed to have multiple services meant to be deployed during the class (inside the classroom) as well as outside of the classroom. The development of mi-clicker occurred during Fall 2009 and early Spring 2010 semesters.

2. Deployment of Educational Application over WiFi Infrastructure: On a voluntary basis, we have given to undergraduate students of the systems programming class, cs241, Google Android phones. On these phones, we have deployed mi-clicker. The deployment and usage of the WiFi phones and mi-clicker application occurred in the second half of the Spring 2010 semester.

3. Feedback from Students: We have asked students through questionnaires about their experiences with the WiFi-phones, WiFi infrastructure and mi-clicker application services. The students provided feedback at the end of Spring 2010 semester when they returned the WiFi phones.

4. WiFi Infrastructure Measurement and Evaluation: We have measured and evaluated the usage patterns and underlying WiFi infrastructure in the classroom where cs241 class met during Spring 2010.

3. Models and Assumptions

Mobile phones have been used for personal usage and are ubiquitous in their usage anywhere and anytime. However, as students enter the age of mobile learning communities, we wanted to ask students how they would envision using phones as educational tools inside and outside of classroom as well as what was their experience with the educational application ‘mi-clicker’ used on WiFi-based phones.

3.1 User Model

The students we have surveyed about educational applications to be used on mobile phones have been undergraduate students in their 2nd and 3rd years of study. We have provided questionnaires to students of the Systems Programming class, cs241 in Fall 2009 to find out the potential educational usage of phones and in Spring 2010 to find out their experience after deploying the ‘mi-clicker’ educational application. The questions in Fall 2009 questionnaire for the potential educational usage have been grouped in three categories: (a) what kind of potential educational usage of phones do the students see inside and outside of classrooms, (b) what kind of potential usage of phones do the students see with respect to better communication with instructor and cloud computing.
teaching assistants, and (c) what kind of potential usage of phones do the students see in group projects and communication with classmates in general.

The most frequent answers to the first category of questions were: (1) advanced quiz application (similar but better i-clicker application) from the phone, (2) notifications about due dates, exams and tests, (3) access to class material on the phone such as access to slides, home page of a class, and (4) access to Wikipedia and dictionary of basic concepts during lectures.

The most frequent answers to the second category of questions (communication tools with instructors and teaching assistants) were: (1) chat application with teaching assistant and instructor during and after lecture. Especially, students saw needs to ask teaching assistants questions during lectures and get quick clarification. The students also would like to give feedback to instructors and let them know if the lecture is going too fast or too slow. (2) phone client software for newsgroups, email, instant messaging, ssh, text editor, gcc to communicate with instructors, teaching assistant and others more efficiently.

The most frequent answers to the third category of questions (group communication tools) were: (1) group notification capabilities about joint meetings, (2) group chat, (3) group access to SVN tools and notification of updates, and (4) group meeting schedules.

3.2 Application ‘mi-clicker’

Based on the feedback from the Fall 2009 questionnaire in cs241 class, we have developed an educational application for undergraduate students, called ‘mi-clicker’ (Mobile Illinois Clicker). ‘mi-clicker’ is a tool-kit for inside of the classroom that includes the following functions: (a) take quiz function, (b) get announcements (MPs, homework, exams) function, (c) text message with teaching assistant and/or instructor, and (d) ask question via audio to teaching assistant and/or instructor.

The mi-clicker basic service architecture is web-based and client-server based, where the student clients are phones. The clients on the desktops will have interface for the teaching assistant/instructor as shown in Figure 2.

![Figure 2: Main Architectural Design of ‘mi-clicker’ Application and its Functions Quiz, Announcement, Text Messaging and Voice Messaging](image)

The server architecture is a web server with (1) extensive database (mySQL) keeping information about the class students that are registered in the particular class, quiz questions, quiz results, files and any other information related to mi-clicker content, (2) PHP interface between the secure database and the core services and management interface, (3) management interface that serves the instructor/teaching assistant to enter quiz questions, announcements, see students answers, get text messages, questions, and provide text for answers, and (4) core services that are responsible for distribution of quizzes, announcements to students, receive answers, receive text/audio questions and store them safely in the database via PHP interface. It is important to stress that the functional part storing student database has been stored on the secure departmental server during the experiment. The students and instructor must authenticate to get access to the server and receive announcements, conduct quizzes and leave text/audio messages for instructors/teaching assistants. The student client system includes interfaces to the four educational services as shown in Figure 3, and indicates the usage of the services. The student mi-clicker services have been implemented using Java on Android G2 phones.
4. Deployment

In Spring 2010, we have deployed G1/G2 Android phones to approximately 100 students of the cs241 class, which took place in a large classroom 1404 SC, seating about 200 students. The participation of the students was voluntary since we wanted to test the robustness of the mi-clicker services on larger scale of students, test the WiFi infrastructure in large classroom 1404 SC, test response of a different group of students to the needs of educational tools on phones, and see the complexity of deploying large scale mobile devices.

We had to face the following issues: (a) logistics to distribute, manage and collect G1/G2 phones to/from the students, (b) distribution, setup, configuration, management of updates, usage of mi-clicker educational software, (c) student assistance via help-desk support infrastructure, (e) participation of students.

4.1 Logistics

The first step to the distribution of phones was to create a lease form since mobile phones were considered educational devices leased to the students by the department for a specific period of time of their educational process. The prior approaches we have seen in other educational institutions were to create a mobile phone lab, with phones hard-wired to desks, for software development, but not for individual mobile usage. However, since we wanted to explore the mobile usage of phones in educational environments, we wanted the students to carry the phones with them all the time, and use them in and out of classroom. Hence, we have developed processes to lease and return a phone for the semester duration. The lease form is attached in Appendix 2.

Further logistic issue to be solved was short term replacement of phones. Our process clearly specified when a phone could have been replaced (e.g., hardware faulty – we had memory damage as well as USB plugs faults). The phones were also replaced without charge in case of theft with a proper police report. Otherwise, in case of missing phones, students had to replace them on their own or be charged in the amount of the phone cost.

4.2 Software Distribution, Configuration, Setup, Management, and Usage

We have setup the mi-clicker server on the departmental server with a secure database that held the names of the cs241 students. The authentication service was setup in such a manner that each student and instructor/teaching assistant had to register through the main MLC mi-clicker website and the system would email the user his/her password. With respect to the mi-clicker application software on the phones, we have used Google Android Market (applet store) and informed students on the mi-clicker website how to download the phone applet. The students could find the instructions at http://mlc.web.cs.illinois.edu/download_page.php. The mi-clicker website also had information how to connect to our WPA wireless protocol infrastructure, usage instructions for instructors, and student service explanation of the buttons on the phone, and help instructions of error messages.

4.3 Students Participation

In the cs241, Spring 2010, we made the student participation voluntary due to the questions about the robustness of the mi-clicker software, accessibility of WiFi infrastructure to many students at the same time and
number of phones for the class (we did not have phones for each student of the class). The participation of students was time limited between April 1 and May 15, 2010.

4.4 Deployment Process

We have deployed approximately 40 G1 and G2 free phones in cs241 class. The students used the phones in class and also for their own personal usage. The G1/G2 phones allowed them to include their own SIM card and so many of them put their own SIM cards into the phones and used their own data and voice service. (Note: As a department we were not allowed to pay for data or voice service from governmental funds, hence all our educational services were running over WiFi infrastructure. However, when students used their personal SIM cards, they could access any other services, e.g., Instant Messaging, email, and other services available on Android phones through Verizon service provider).

The students used the mi-clicker phone in class for quizzes and announcements. The text messaging was used between the students and the teaching assistant. The voice messaging was not used. The students used the phones extensively for their personal usage outside of the classroom. Many difficulties came up using ‘mi-clicker’ software as well as phones within the classroom. The challenges and difficulties can be summarized as follows: (a) difficulty with the wireless infrastructure inside of the classroom when large number of students aim to access resources at the same time, (b) issues with the authentication service, (c) unclear understanding from the instructor and students sides how to include usage of phones into the lecture and educational process, and (d) difficult interfaces to use.

4.5 Students Feedback – ‘Exit’ Questionnaire

At the end of the cs241/Spring 2010 semester, we have asked students who agreed to participate in the Android/mi-clicker experiment to fill out the “Exit” questionnaire when they return the phones. 19 students responded to our “exit” questions. We have asked students questions in three categories: (a) what were students’ experience with the mi-clicker software, (b) what kind of problems did students experienced with the phone, (c) what kind of future usage of phones/software do the students envision in an educational environment.

The most frequent answers for the first category were: (1) class did not use mi-clicker often enough, (2) texting with teaching assistant was useful, (3) quiz when used was very useful, and (4) mi-clicker did not integrate well with the classroom environment and the cs241 teaching style.

The most frequent answers for the second category were: (1) login, password and authentication issues since it took very long time to get authenticated and even once one got authenticated, due to short in-activity, wireless connections were lost and new re-authentication was needed; (2) device power issues since the G1/G2 devices (especially G1 devices) have very short device life-time; (3) WiFi availability issues where the wireless infrastructure availability proved to be unstable and access was not always possible; and (4) slow access to web-server since when students submitted quiz answers or requested a list of announcements or aimed in any other way to communicate with the web-server, the access was very slow which again does not present viable solution for time-critical educational applications such as quizzes, messaging, and other educational tasks.

The most frequent answers to the third category of questions were: (1) provide stable WiFi infrastructure for future mobile applications; (2) teach Android programming in various classes (3) provide simple and stable educational applications in large classes, and (4) provide access to general tools such as email, newsgroups, web-browser, ssh and other general tools outside of classroom over WiFi.

5. Evaluation of Infrastructure and Software

In this section we will investigate the readiness of our communication infrastructure and consider two major aspects: (a) what are the HTTP request quality measurements under the WPA Enterprise networks (since our mi-clicker application tool-kit is web-based and concerns have been expressed about the slow access to the web-server), and (b) what are the WiFi conditions in the classroom 1404 SC (since most of the student concerns included WiFi availability in the 1404 SC classroom). The first aspect will be measured with respect to the success ratio of HTTP requests to the web-server, RTT (Round-Trip-Time) and available bandwidth. The second aspect will be measured under different scale of phones, different topologies, different WiFi authentication protocols, different power provision, with heterogeneous devices, and with goals of finding dead spots.
We have conducted large-scale Android phone experiment during the Spring Break of 2010 in the Siebel Center, room 1404 which has 200 seats. The reason for the time frame was that the classroom was available to us through the whole week and so we could conduct multiple and diverse controlled experiments. The numbers of phones in the experiments were 80+ G2 phones and 40+ G1 phones. We have conducted two types experiments: (1) large scale experiments with phones only (Pure Large Scale), and (2) large scale experiments with mixture of phones and laptops (Mixture Large Scale). In the first experiment type we have included 120+ phones and in the second experiment type we have included 20 or 80 G2 phones plus several laptops.

5.1 HTTP Request Quality Results

To obtain ‘mi-clicker’-like HTTP request quality results, we were running test scripts between Android phones and a simple web-server to stress-test the HTTP protocols and the access to the web-server over WPA enterprise network. We have tested the same HTTP test-scripts in the 1404 SC classroom as well as at home, and we tested the HTTP requests on phones and on laptops.

The experimental results showed a long RTT delay for a single HTTP request. At Siebel Center, the RTT delay was 4-8 seconds for a phone HTTP request versus less than 1 second in a home environment. At Siebel Center, the RTT delay was 4-8 seconds for a phone HTTP request to the web-server versus less than 1 second for a laptop. Hence, the RTT delays clearly indicated that the HTTP request delays were unacceptable for interactive applications such as mi-clicker and changes need to be considered.

Another metric we have measured and that was the failure ratio of HTTP requests to the web-server. Our results showed that HTTP requests on phones showed very high failure ratio up to 37-64%, which was again not acceptable for interactive applications such as quizzes in mi-clicker. Table 2 summarizes the results.

<table>
<thead>
<tr>
<th>Metrics/Results</th>
<th>Siebel Center/Phones</th>
<th>Home/Phones</th>
<th>Siebel Center/Laptop</th>
</tr>
</thead>
<tbody>
<tr>
<td>RTT (Round Trip Time) of single HTTP request</td>
<td>4-8 Seconds</td>
<td>&lt; 1 second</td>
<td>&lt; 1 second</td>
</tr>
<tr>
<td>Failure Ratio of HTTP requests</td>
<td>37-64%</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

Table 2: Results of HTTP Requests to Web-Server over WPA Enterprise Network

5.2 WiFi-Related Results

Based on our overall Spring 2010 measurements, we have concluded that (a) WiFi stability was low in the 1404 SC classroom, and WiFi (re-) association/handoff was a big issue; (b) G1 phones performed worse in terms of WiFi connectivity than G2 phones. Hence, after this finding, we have done most of the WiFi experiments with 80+ G2 phones; (c) WPA Enterprise networks had longer delays than UIUCnet due to authentication, encryption, decryption overheads in the range of 0.2-0.6 seconds. UIUCnet does not have encryption; (d) SC 1404 classroom had insufficient deployment of access points (APs). The anticipation and information from the system administration was to have four APs for the SC 1404 classroom, but after our measurements, we found only two APs served this large classroom. We have found dead spots in the middle of the classroom (very low SNR signal), and with large number of phones, we experienced congestion which increased the end-to-end delay dramatically, with RTT up to 4-6 seconds per HTTP request. Below we present in detail our measurements of the WiFi-related infrastructure results.

In the Experiment 1 we had deployed maximum number of phones 80+ G2 phones and 40+ G1 phones in SC 1404 classroom with random phone placement and using WPA2 enterprise network for 1hour. We have also done random assignment for TTLS (Tunneled Transport Layer Security)/PEAP (Protected Extensive Authentication Protocol). All phones have been connected to power plug (we have brought into the classroom power extension courts to connect all phones to power jacks which are not available during regular classroom times.). Under this experimental setup, we had 26 hosts respond with the HTTP request. The failure ratio was around 64% and delays to access the web-server were approximately 4.5 seconds to a regular website and 6 seconds to the MLC mi-clicker website. Results are shown in Table 3. (Note: In this experiment, even though 120 phones were deployed, only 26 phones reported their results back to the database during 1 hour period. This was due to the fact that we had to manually start the Mi-Clicker applications in every phone, ensuring WiFi is connected when the app starts. The manual starting process for all the phones took more than half an hour.)
<table>
<thead>
<tr>
<th></th>
<th>TTLS</th>
<th>PEAP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Failure Ratio</td>
<td>0.6527</td>
<td>0.6349</td>
</tr>
<tr>
<td>Delay(web) in seconds</td>
<td>4.4940</td>
<td>4.5240</td>
</tr>
<tr>
<td>Delay (mi-clicker) in seconds</td>
<td>5.8872</td>
<td>6.1212</td>
</tr>
</tbody>
</table>

Table 3: Experimental Results under Experiment 1

In the Experiment 2, we have removed the G1 phones due to their low WiFi performance, and considered 80+ G2 phones with laptop interference. We have considered random placement of phones and laptops were adjacent to phones with streaming video to stress the interference effect. We have used the WPA2 enterprise network for 1 hour. We have employed random assignment for TTLS/PEAP. Under this experimental setup, we had 52 nodes respond, the failure ratio of HTTP requests was approximately 50% and the delay to a regular website was approximately 4.1 seconds and to the MLC mi-clicker web-server approximately 5.47 seconds. Results are shown in Table 4.

<table>
<thead>
<tr>
<th></th>
<th>TTLS</th>
<th>PEAP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Failure Ratio</td>
<td>0.4524</td>
<td>0.5667</td>
</tr>
<tr>
<td>Delay (web) in seconds</td>
<td>4.1727</td>
<td>4.0887</td>
</tr>
<tr>
<td>Delay (mi-clicker) in seconds</td>
<td>5.4338</td>
<td>5.5254</td>
</tr>
</tbody>
</table>

Table 4: Experimental Results under Experiment 2

The HTTP request times to web-servers have been large in Experiment 1 and 2, and hence we wanted to test how a HTTP request will perform under UIUCnet network (Experiment 3). Hence, we have deployed 20 G2 phones with random placement of phones. We have tested the phones under UIUCnet for 1 hour and the phones have been without power plug. In this case, on average 9 hosts responded out of 20, hence we achieved failure ratio of 0.3748 with delays of 1.8724 seconds HTTP request to a regular website and 2.3630 seconds HTTP request to the MLC mi-clicker website.

We have altered the Experiment 3 with adding more phones and laptops in Experiment 4. We have experimented with 80+ G2 phones, random placement of phones and placed adjacent to them laptops for interference. The HTTP requests were running over UIUCnet networks for 1 hour. At this point, on average 40 hosts responded, yielding failure ratio of 0.4843 and delays of 3.5941 seconds for regular website and 5.3375 seconds for MLC mi-clicker website.

5.3 Possible Reasons for Long Delays

As we have discussed above, having 4-8 seconds access to the MLC web server is unacceptable for the interactive educational applications such as mi-clicker. Hence, we have analyzed some of the possible reasons for the long delays to take next steps for future generations of MLC educational tools. We have found several problems with the software and the phone themselves (without laptops):
1. mi-clicker implementation since it used non-persistent HTTP connections, contributing on every HTTP request to additional delays due to re-association of connections.
2. Login and password issues since mi-clicker could not cache passwords, so on every loss of WiFi and/or server connection, password had to be retyped, causing additional delays.
3. Authentication process was a problem because using WPA2 enterprise networks infrastructure induced additional authentication and security overheads.
4. Congestion was a problem because we had only two access points available in 1404 SC classroom, hence congestion and bandwidth availability represented difficulties when deploying 100+ phones.

In addition as we added laptops into the mixture (Experiments 2 and 4), which is an expected setup in a classroom since students will bring with them not only phones but also their laptops, we saw additional problems:
5. Laptop interference was an issue since laptops transmitted with larger power than cell-phones, preempting the cell-phone transmission for several rounds, causing phones additional delays with HTTP requests. This is a problem that represents one aspect of the well-known hidden terminal problem.
6. AP preferred treatment of laptops meant that access points gave better chance to laptops to decode laptops’ packets than phones’ packets when they sent packets simultaneously.
7. Phones reactions to interference caused that when phones saw packet drop, their protocol backed off, causing unfairness in the access to the wireless channel between phone and laptop.
Another cause of the large web access delay may have been that the phones’ wifi interfaces adopted PSM (Power Save Mode) instead of CAM (Constantly Awake Mode). The 802.11 standard supports multiple modes of operation, with different power consumption for each mode. When CAM is adopted, the WiFi device stays awake, ready to send and receive packets all the time, hence consumes most energy. On the other hand, when PSM (Power Save Mode) is adopted, the WiFi device goes to sleep, i.e. it stops receiving packets when it has no packet to transmit, and wakes up periodically for beacons. If the beacon indicates there are packets for it buffering at the AP (access point), the wifi device sends a separate PS-POLL message to receive each buffered packet. The 802.11 Access Point (AP) supports PSM by 1) buffering incoming packets for WiFi clients in PSM, 2) indicating the presence of buffered packets via fields in beacon messages, and 3) delivering the buffered packets after the client notifies the AP it’s ready to receive one or more packets.

While the static PSM approach allows the device to save power by only waking up when packets are outstanding at the AP, it introduces extra latency involved in receiving packets via PS-POLL and such delay has been found to be high for interactive applications such as web browsing (Krashinsky et al. 2002). The APs in the WLANs of UIUC set default beacon interval value to be 100ms, therefore each packet delivered to the phone from the AP has extra delay caused by buffering at the AP which is up to 100ms. Since HTTP connection takes several rounds of handshake to build, the phones may take 400-500ms to build a HTTP connection. In order to mitigate this problem, many WiFi devices today also implement a technique known as adaptive PSM, where the device switches between PSM and CAM based on some heuristics.

According to our observation, the G1, G2, and droid phones that we used in the experiments all adopted adaptive PSM, and the switch between PSM and CAM was triggered by reception of a threshold number (denoted as TH) of packets or lack of network activity for a pre-defined duration. The device notified the AP of its transitioning to PSM or CAM by sending NULL data frames with the power management bit (in the MAC header) set to 1 or 0, respectively. However, we found out that the default value of TH on the G1, G2, and droid phones was too large, so that the traffic flow of our python program was not large enough to trigger the phone operating at CAM mode. Therefore in our experiments, the phones still encountered large delays caused by PSM, while the laptops which always operate at CAM mode worked better.

We reset the TH on the droid phones to a smaller value and observed significant delay decrease, while at the same time, the power consumption also increased. However, we found out that this solution did not work on the G1/G2 phones’ WiFi interfaces. When we reset the TH to a G1/G2 phone smaller value, although the phone actually operated at CAM mode when the traffic flow was large (we observed so using a WiFi sniffer), it set the power management bit (in the MAC header) to 1 (indicating PSM mode) in EVERY single packet it sent out. Therefore, the AP assumed that it operated at PSM mode and buffered all packets. We suspect that the discrepancy between the power management bit value and the actual operation mod was a bug in the NIC driver of G1/G2 phones.

Sometimes the software tools we used to build our program could cause the delay. For example, our test program was running using ASE, when we upgraded ASE to a newer version, the HTTP request delay decreased by 40ms.

Besides problems with the phone software mi-clicker implementation, phone configuration, laptop interference, we anticipate that the delays we saw with HTTP requests between phones and web-servers might also be due to using the Meru network devices:
8. Centralized Scheduling - The Meru networks use centralized packet scheduling algorithms over the whole WLAN. Hence, this can influence the airtime fairness, seamless handoff, and overall scheduling time
9. Laptop Access Times – We have done laptop measurements in the Siebel Center (in promiscuous mode) and it happened frequently that our laptop received 0 packets from the access point for over 2 seconds. This is again very undesirable if we consider interactive learning applications in classrooms. We suspect that the Meru AP stopped serving clients when it processed handoff or other scheduling, and this processing time could be as long as several seconds.

6. Related Work

Advances in wireless technologies have allowed users to communicate ubiquitously anywhere and anytime and made it possible for users to access and exchange information through wireless handheld devices such as cell phones, PDAs, tablets, and other wireless devices (Lyytinen et al 2002). We are seeing an explosion of wireless communities and applications (e.g., Buddy Finder, Proximity Dating, and Cab Ordering) as wireless
Infrastructures and deployments in wireless digital cities and wireless local communities emerge pervasively. For example, in Helsinki, Finland, people discuss, plan, and manage local events with cell phones through the Helsinki Virtual Village (Helsinki 2008). On the other hand, wireless local communities are fixed to a specific area, e.g., a shopping center or tourist park (Sun 2007), to share their local experiences with each other.

It is important to stress that these wireless communities are very different from traditional Internet-based online communities, in which people are connected with one another but neither know each other in person nor necessarily care where they are at any specific moment. More specifically, although people in different places are connected through Internet-based communities, information exchange alone is not likely to lead to a sense of social interaction close to what happens in the real-world communities (Sprout et al. 2004). In real-world communities, social interactions occur in physical and social contexts that are shared by those involved. However, most contextual cues are filtered out in text-based communication through the Internet. That is why face-to-face communication has usually been found to be more effective than computer-mediated communications at fostering community (Whittaker 2003).

In contrast, in wireless communities, people are much more closely bound to each other, through a sense of sharing a common physical and/or social context. In such communities, for example, it is possible for members to access the contextual cues directly or with the help of information technologies. For mediated communications, research has found that joint attention and social linkage are necessary conditions for effective information exchange (Clark et al. 1981; Nardi et al. 2002). As a result, a sense of sharing the same physical and/or social context helps to bind people more closely in wireless communities, leading to “contextual communality.”

The realization of mobile learning communities will also depend on prior work in the space of (1) active participation of members (Whittaker et al. 1997), (2) the design and development of advanced mobile application services for wireless communities/users that will facilitate the sharing of contextual cues, (3) assurance of stable underlying software and hardware infrastructure (i.e., reliability, robustness, privacy, and security of services in the mobile educational environments), and (4) assurance of social trust from the various users.

7. Final Conclusions

Over the one year, we have explored various dimensions of mobile learning communities in our department at the University of Illinois. The basic hypothesis is that our universities will have students who carry phones (not only laptops) and they will create mobile learning communities during their time at our campuses. So we need to think about if we as formal educational institutions are ready for these mobile learning communities. The one year study aimed to answer one of the many questions: how is our infrastructure ready for this large scale deployment. The answers are complex and include large number of considerations before we respond with the final answer “Yes, we are ready for phone-based mobile learning communities”. As we showed, some aspects of the MLC educational process can be mastered within short time period such as establishing logistics of the phone deployment, but some aspects of the MLC educational process will take several iterations to provide stable MLC environment for our students and instructors. So at this point, our answer is “We are not quite yet there.” However, we believe that the deployment of phones as educational tools in and out of classrooms should continue. As we experiment with phones and other mobile devices, and increase the number of students who can program new educational and other useful applications, the mobile learning community at Illinois becomes stronger, giving strong advantage to our students in their future.

References

Acknowledgement

This effort has been funded by the National Science Foundation (NSF) CNS IT grant NSF 0855129, and industrial sponsors such as Vodafone Educational Grant, Qualcomm and Google. The opinions presented in this document are our own and do not represent the opinion of the funding agencies and sponsors. We would like to thank the MLC professors R. Campbell, I. Gupta, R. Kravets, W. Sanders, S. Poole, L. Angrave, S. Kamin, and mi-clicker undergraduate developers: Alex Loeb, Michael Ford, Craig McIntyre, Ohm Patel, Samuel Retika, Ang Pan, and the mi-clicker testers: B. Gass, W. Jang, M. K. Kang, Y. P. Malimas, A. Rachwalski. We would also like to thank students of the class cs241 in Spring 2010 and cs425 in Fall 2009 at UIUC for their cooperation in the MLC project.