UbiConn: Providing a Ubiquitous Connectivity Experience

CSE 561 Project (Autumn 2004)

Katherine Everitt, T. Scott Saponas, Susumu Harada
University of Washington
{everitt, ssaponas, harada} @ cs.washington.edu

ABSTRACT

Despite the proliferation of mobile devices, mobile connectivity is still lacking. Information access is a powerful tool in wired environments. It is even more important for mobile devices, which have limited storage capacity and are not ideal for primary information storage as they have the potential to be lost or broken. The combination of connectivity and mobility will enable users to access any data from anywhere. We present a model of multi-channel proactive connectivity that supports this goal. Our two-pronged approach includes a proactive caching mechanism to support areas of low speed or no coverage, and we discuss the potential for a multi-channel adapter to prevent unnecessarily dropped connections.

INTRODUCTION

Background

The proliferation of mobile devices along with wireless networking capabilities over the past several years has led to an increase in demand for greater network coverage and continuous connectivity. Cell phones have evolved from being a simple telephony device to a powerful information exchange system; laptops and PDAs with Wi-Fi capabilities are becoming more common in offices and homes and even in public places such as cafes and libraries.

	Throughput	Latency (ms)
Wi-Fi	56 Mbps	<1
UMTS	300 Kbps	250
EDGE	100 Kbps	450
GPRS	40 Kbps	650

Table 1: Comparison of existing wireless network technologies.

Many technologies and standards are emerging, such as 802.11 variants and GSM (Global System Mobile) based technologies such as GPRS (General Packet Radio Service), EDGE (Enhanced Data rates for GSM Evolution), and UMTS (Universal Mobile Telecommunications System). However, the current state is such that there is no integrated solution that allows a user to simultaneously take advantage of multiple wireless technologies depending on which are available at a particular moment. The user is required to make a tradeoff between connection speed and reliability. Today's wireless networks can be largely divided into two main categories: those that provide wide area of coverage and thus relatively continuous connectivity yet offer low bandwidth (e.g. GPRS), versus those that offer high bandwidth while usually remaining limited to a small area (Wi-Fi). Taking the union of these two classes of connectivity will allow the user to remain connected all the time and have access to broadband connections.

There have been a number of related attempts at providing an integrated wireless connectivity service at various levels. They are mentioned in our related work section, but all of them either involve a significant modification in the underlying network infrastructure, or do not offer a way to integrate across different network technologies simultaneously. Thus, they do not allow the user to make use of multiple available network connections at the same time. We also list in our related work section several emerging technologies that seek to fill the gap between ubiquitous availability and high bandwidth, none of which are enough to eliminate the need for an integrated system as described above.

Despite the absence of systems offering seamless connectivity across different wireless network technologies, there are certain users out there who have the knowledge and the resources to configure or build their system in a way that provides them with as close to a ubiquitous connectivity experience as possible using the existing network technologies. We call this type of user the "savvy user" and believe their techniques provide useful insight into what functionality a UbiConn system should provide. For example, such a savvy user may anticipate that she will be moving to a location with no network coverage and thus preemptively download web pages or emails that she might wish to have access to while offline. She may also manually switch from one network connection to another, for example from Wi-Fi to GPRS, when she anticipates that one will no longer be available. However, these coping mechanisms require the savvy user to predict what information she will need and take the time to access it in advance. This takes time and effort on the part of the user, and unsophisticated users will not be able to do even this much.

Goal

Due to the large number of wireless network technologies and their uses, we chose to look at a specific but representative subset. For this project, we focus on a particular use case scenario involving an average mobile user with a laptop computer capable of two of the more common wireless connections, namely GPRS and Wi-Fi. The user roams from one location to another with different network coverage. By "average user" we assume that the user is not a savvy user and does not have the knowledge to manually configure her system to deal with changing network connectivity.

To get an idea of what such a user may experience, we conducted an initial analysis of the network connection management scheme on a typical mobile device setting. We used a laptop computer running the Windows XP operating system and equipped with a built-in Wi-Fi card and a GPRS card. We took the system through areas of varying wireless network coverage, and observed the following behavior with respect to three main types of applications. The motivation for the choice of these application types is described in the Use Case Scenario section.

When transitioning from the office with Wi-Fi access to a bus with only GPRS connection, the web browser continued to work over the GPRS connection while both instant messenger (IM) and telnet applications initially disconnected the user. As the IM application was set to attempt an auto-reconnect, after a few seconds it reestablished connection using the GPRS connection, while the telnet application required a manual re-connect. When Wi-Fi access is made available again at the café, the computer detected the presence of a Wi-Fi network but failed to establish a connection to the Internet. After several manual rescans on the Wi-Fi adapter, the computer was able to reestablish connection through Wi-Fi. In the process, IM and telnet connections were disconnected, and after a minute or so, reestablished connection using the Wi-Fi connection. The web browser automatically started using the Wi-Fi connection as soon as it was available.

Our goal is to create an integrated system that will provide an average user with a UbiConn "experience" in today's environment where multiple disparate wireless network connections are available, without the need for any modification in the network infrastructure. We first establish a use case scenario surrounding this average user and a set of core applications that she will be using, and establish certain quality of service expectations that she has in dealing with these applications. We then analyze how well they are satisfied under a base setup of a laptop with GPRS and Wi-Fi cards configured to connect to their respective networks whenever they are available. We then compare this result to our system that provides automatic pre-caching of web content and semi-transparent application-based routing (with the option for the user to intervene and specify desired routing behavior) of network packets to appropriate network connections.

RELATED WORK

The areas of work related to this project fall into the following categories: wireless technologies, combinations of wireless technologies, and caching relevant data to provide the illusion of connectivity.

Technologies

There are several emerging wireless technologies that promise greater range and bandwidth. HSDPA (High Speed Downlink Packet Access) is an enhancement to UMTS that is expected to provide throughput rates of 1 Mbps on average, but is not slated to be deployed until 2006, and is still significantly slower compared to Wi-Fi [14]. The IEEE 802.16 standard commonly referred to as Wi-Max seeks to provide wireless access of up to 75 Mbps over a range of several miles [15]. However, it is primarily designed for providing last-mile broadband connections into businesses and residences, and the mobile version of the standard is not expected to offer the same level of flexibility offered by the GSM-based technologies.

There have been several previous proposals for integrating UMTS connections and Wi-Fi 802.11 wireless LAN networks [1]. Wireless and cell phone companies such as TMobile are providing 802.11 and GPRS integrated solutions under one service provider. Other companies are going the other direction and providing voice solutions over Wi-Fi [6]. These various industrial efforts to combine multiple wireless technologies are all based on the mobile device only using one connection medium at a time. We seek to go beyond this and provide different qualities of service to different applications simultaneously using multiple wireless connections concurrently.

Heterogeneous Networks

The BARWAN (Bay Area Research Wireless Access Network) project [2] enables a mobile client to transition amongst heterogeneous wireless technologies ranging from infrared networks to Wi-Fi to Regional-Area Satellite Data Networks. Their focus is on providing seamless vertical handoffs across network of different wireless technology and coverage profiles. They use an implementation of a Mobile IP variant as well as specific coverage geography information to facilitate connection handoff from one coverage area to another. While BARWAN deals with vertical handoffs between networks, our system uses heterogeneous networks in parallel to choose the best network for each application. We use available resources of connections, power, and disk storage space to improve the experience and maintain connectivity by using every available network in parallel. This helps us select the best network for the service. For example, low bandwidth high connectivity applications such as telnet can be serviced by GPRS. This will prevent it from being significantly affected by changes in the network connectivity, as switching unnecessarily between GPRS and Wi-Fi simply because Wi-Fi is faster will interrupt the telnet session. Thus, we use both GPRS and Wi-Fi, if they are available, to maximize what is most important for the application: high bandwidth or continuous connectivity. Our approach also differs from the BARWAN solution in that we do not impose any changes to the underlying network protocol or infrastructure.

Homogenous Networks

The CHOICE network [8] deals with transitioning between networks of the same type. Using the CHOICE network, a mobile client is able to transition across multiple Wi-Fi hotspots offered by different service providers. CHOICE manages the authentication and account management aspect of these handovers. However, it does not deal with multiple network technologies, and thus cannot benefit from the advantages of a network with more coverage such as GPRS. For the CHOICE network to maintain connectivity, it requires overlapping hotspots.

Caching Systems

Coda [18] is a distributed file system that supports disconnected operation. It uses cached data to improve data availability when the servers are not accessible. Coda caches some content locally. In a disconnected state this cache is used to serve file requests. It uses whole-file caching, which makes sure an entire file is always cached, to cause a simpler failure model. The Coda system has a priority-based cache management system with both implicit (based on recent reference history) and explicit (based on user requests) sources of information for the decision of what to cache. It uses both user specified priority and recent usage to determine ranking. Our caching system works over a different domain; web pages rather than files. Our caching heuristics are similar to this model, supporting both user-specified requests and implicit measures. Our system has more discrimination between implicit content than Coda, as it automatically ranks different pages according to our heuristics. We cache HTML pages in a piecewise manner, preferring high-value low storage cost content such as web text over images.

AvantGo [7] is a commercial service for portable devices that does caching of content while the user syncs with the server. It allows customers to sync with the server at the base station or over the Wi-Fi. It automatically downloads specially formatted websites, such as www.cnn.com, while the connection is available. It provides high-level ways to select sections of the web, and also allows users to define their own custom downloads. This allows for the caching of data but requires the user to specify what websites she would like to have cached. It does not do the updating automatically, nor does it automatically determine what the user may need.

Prefetching [9,10] is a proactive method of caching which attempts to predict what the user will need and fetch content before the user requests it. As opposed to short term prefetching, which uses recent access rates to select cached content, long term prefetching uses long-term steady-state access rates to identify objects to cache. Long term prefetching uses statistical models to predict the likelihood of a page being accessed, and caches pages based on this metric. Click-ahead caching refers to proactively caching links one page further than the current page, speeding up response times when those links are accessed.

Web browsers such as MS Internet Explorer cache content as it is accessed. They provide support for an “offline mode”, which allows users to access some of the content they have seen already. However, this content is not updated unless the user views it again, and it only stores pages that the user has viewed, not those he is likely to view.

Programming for Mobile Systems

Rover [3] is a toolkit for mobile applications that enables the development of mobile transparent or mobile aware applications. Mobile aware applications are given information about the network connectivity status in order to make decisions. Mobile transparent applications do not use this information. The primary function of this toolkit is to provide a new paradigm of programming interfaces for mobile computing. Their main tool for mobile computing is having applications use distributed objects as their main method of transferring data over the network. This allows their toolkit to manage network usage by many applications over slow connections and queue up future transfers when in a disconnected state. Our approach is to work with mobile transparent applications but support them so that they can behave more like mobile aware applications. We accomplish this by guessing what type of network service an application needs. Rover also provides a “Rover HTTP proxy” to support web browsers transparently. To the best of our knowledge, this proxy system does one time caching plus prefetching. This means the Rover HTTP proxy tries to cache any web page it hasn’t cached before and also caches a user specified number of sub-pages deep. Our caching system contains this functionality but also does predicative caching and proactive refresh of the cache.

One possible solution to the problem of maintaining connectivity as the mobile device is transported from one network to another is to use a Mobile IP [4, 17] solution. Mobile IP is a proposed standard augmenting the existing version of IP that seeks to address the problem of maintaining TCP connection status when a mobile device transitions to a different network (and thus changes its IP address). Mobile IP works by associating the mobile node with two IP addresses, the home address and the care-of address. The home agent at the home address redirects traffic to the up-to-date care-of address, which then forwards it on to the mobile node. Mobile IP encounters challenges when faced with security implementations such as firewalls and ingress filtering. Mobile IP could be used to maintain connectivity when handing off between networks. However, it is a more complicated scheme that requires a home agent and incurs the overhead of updating the care-of address settings each time the mobile node is moved. For our project, we place our emphasis on providing ubiquitous connectivity across heterogeneous networks, and thus view Mobile IP as a possible technique for solving the TCP connection persistence within each network, but not as the main focus of our pursuit.

Load Balancing and Multi-Channel Wireless

In the context of the MultiAdapter, there are existing solutions that provide load-balancing functionality over multiple network interfaces on one computer. For example, Adaptive Load Balancing technology from Intel allows a single server with multiple network adapters connected to the same network to multiplex its transmissions across all the adapters [16]. There are also wireless broadband routers (such as Linksys Wireless-G Broadband Router with SpeedBooster and X-Micro WLAN 11g Turbo Mode Broadband router) that provide the ability to load balance over multiple broadband connections (e.g. DSL and cable modem) as well as allowing a wireless client to multiplex transmission over multiple channels to achieve increased throughput. However, these solutions only work over one type of network medium (e.g. wired Ethernet or 802.11) and do not offer intelligent routing of traffic over network interfaces of different mediums based on flow-specific filtering.

Our contribution lies at the intersection of this set of related work. We provide for concurrent access to heterogeneous networks, and have a caching system for areas of no connectivity.

USE CASE SCENARIO

In considering obstacles face a non-savvy mobile user who wants ubiquitous connectivity, we developed a hypothetical user and a use case scenario around her in order to identify the problem points and constraints. Following is a description of the use case scenario.

Our user owns a laptop computer equipped with a Wi-Fi card and a GPRS card, and works at an office that has building-wide Wi-Fi coverage. We identify three core applications for our user that are meant to be a representative subset of common applications that require network connectivity. They are: 1) web browser, 2) instant messaging (IM) application, and 3) remote terminal application (e.g. telnet). Her primary use of the web browser is for accessing news websites. She uses her instant messaging application often to keep in touch with colleagues, so it is desirable that her IM connection be available as much as possible. She also uses her remote terminal application for accessing her corporate server to run tasks, so it is highly desirable that this connection remains uninterrupted. Summarizing the different applications our user uses, they can be categorized into the following respective groups: 1) high bandwidth desirable but unnecessary and availability of content highly desirable, 2) low bandwidth sufficient but connectivity desirable, and 3) low bandwidth sufficient and connectivity highly desirable.

The following paragraph describes the sequence of actions that our user follows in our scenario. First, she starts off in her office where she has all three of the applications running using the Wi-Fi connection. She then takes her laptop with her, leaves the building to get on a bus. On the bus, she desires to continue her work as much as possible given the available network resources. During the bus ride, she has access to the GPRS network but not to any Wi-Fi network. She then disembarks at a café that has Wi-Fi access, and resumes her work. After some time, she heads down to a subway station to board a train to her client's site. During the subway ride, she does not have access to Wi-Fi or GPRS. Once she arrives at her client site, she regains both her Wi-Fi and GPRS access.

SYSTEM DESIGN

As our use case scenario reveals, there are two major facets to providing a UbiConn "experience": connection sustainability and data availability. In order to address these two needs, our system is composed of two modules, namely the MultiAdapter and the Caching System.

MultiAdapter

In order to enable certain subset of applications to use the GPRS connection while others to use the Wi-Fi connection when they are available, we developed a network filter we call the MultiAdapter. The MultiAdaptor intercepts network traffic from applications and routes the traffic to the appropriate connection based on the application type (or protocol type). The decision about which connection to route the traffic to is based on several factors. The most obvious constraint is the availability of each network. If GPRS is the only connection available, then applications that desire a Wi-Fi connection will have no choice but to use the GPRS connection. The second factor is based on the application's quality of service requirements, as characterized in the use case scenario section. Traffic for remote terminal application is routed through the GPRS connection even when Wi-Fi is available, in order to ensure that the connectivity is not interrupted when the user moves to an area without Wi-Fi coverage.

Caching System

Although it is true that wireless network coverage, especially GPRS, is becoming more pervasive, there are still many locations even within a metropolitan area where there is no coverage from GPRS or Wi-Fi. Under such conditions, we can provide the user with a limited sense of a UbiConn experience through caching of web traffic and proactive fetching during periods of available connectivity.

We support two types of caching for different reasons. Click-ahead caching automatically downloads the links from the page a user is already on, which speeds up the response time when the users clicks on a link. Proactive caching, which we implemented to support weak or no connectivity areas, makes predictions about future behavior based on past behavior. In short, we attempt to have available the pages and data the user accesses often.

We would like to support network connectivity for communication, information gathering, general browsing, searching and transacting. Some of these lend themselves to caching better than others. For example, if one accesses the same static HTML news sites routinely, one can predictably cache those sites. Also, communication in the form of email lends itself well to caching and periodically updating, while continuous communication such as instant messenger does not. Accordingly, for this project, we focus our caching efforts on web pages.

There are several factors involved in determining caching heuristics to make sure the user has the data she needs. The type of data can affect the need for it to be updated. For example, news, weather, and stock tickers are updated more frequently than how-to pages or reference documents. Related to this is how often the data is updated, and whether the information is time critical. Data that is very important or user-specific is more important to cache correctly and frequently. User behavior (for example, if they frequently access the same content at a specific time of day) can provide clues to correct cache planning. Page size, content type and download cost should also be taken into consideration when developing caching heuristics.

We developed the following initial heuristics to prioritize cached resources: On a scale of 1 to 5, where 5 is cached 5 times more frequently than 1.

1. Images

   Images are usually more expensive to download than text, and many images such as page headers do not change frequently. Thus, they are updated to the cache less often.

2. Ordinary content

   Most content will fall into this category.

3. Documents that contain the term "news"

   This can later be replaced or augmented with a metric that looks at page structure or samples the frequency of change.

4. Frequently accessed content

   Currently hard coded, but will soon be automatically recorded.

5. User specified pages

It is important to note that content that is not cached is still downloaded when the user requests it. However, it is less likely to be automatically updated, and might be slower in regions of lower-speed connectivity. These worked quite well at classifying web pages in an informal evaluation conducted on the experimenter's web browsing habits. For example, a page from Newsweek was correctly classified as news even though it was accessed from a non-news link.

Implementation

To simplify the implementation of our prototype we make the assumption that the only applications that want the fastest connection (as opposed to most consistent connection) and need caching of data are web browsers. This allows us to build both the MultiAdapter and the Caching System on top of a local HTTP Proxy. The only requirement we have for web browsers is that they support a proxy configuration (which all the popular web browsers do). A user or installer then just configures the web browser to use a proxy on the localhost at a well-known port.

We wanted total control over the HTTP content being proxied, so we built our own HTTP proxy from scratch instead of using a preexisting HTTP proxy. We implemented the HTTP proxy such that when a network connection exists the caching system is sent a copy of all the outgoing HTTP requests so that it may figure out what to cache. When there is no network connection the proxy actually sends the request to the caching system to fulfill the request. In order for the MultiAdapter to know what traffic comes from a web browser we need the proxy to tag it as such. This is implemented using a well-known port. All traffic going out the proxy on port 80 gets changed to a special port number signaling it came from the proxy. Then the MultiAdapter changes its destination port back to 80 and makes note this flow came from the web browser. When traffic comes in to the MultiAdapter, it checks whether it is bound for a web browser and if so changes the source port (it is the source port now because it is incoming data) to the special port number. Notice this scheme only marks web traffic on port 80. This is a limitation of our system.

The MultiAdapter serves two main functions: 1) monitoring network connectivity and 2) routing network traffic to and from applications through the appropriate network adapter. We query the status of the network adapters using the NDIS (Network Driver Interface Specification) interface, allowing us to determine whether the adapter is connected or not and what their link speeds are. The MultiAdapter also acts as a packet filter that monitors traffic passing through each of the network adapters on the machine (in our case the Wi-Fi and the GPRS adapters). This traffic is analyzed, as discussed above, to determine which application they are coming from, and are routed either through the GPRS connection or the Wi-Fi connection, based on the network availability and application connection preference profile, as described in the System Design section.

We implemented the MultiAdapter as a Windows network driver that installs in the same manner as a regular network adapter for a physical Network Interface Card (NIC). The architectural diagram of the MultiAdapter is shown in

Figure 1. The applications communicate directly with one of the protocols in the protocol layer, in our case this is primarily the TCP/IP protocol adapter. Normally, the protocol adapter will communicate directly with one of the underlying miniport adapters, which represent interfaces to the actual network interface cards (NICs). If there are multiple compatible NICs for a particular protocol (e.g. a WiFi card and a GRPS card), in the case of the Windows XP operating system, a routing table manager for the IP protocol automatically routes data through the miniport corresponding to the NIC that is connected and has the highest interface metric (faster NICs receiving a higher metric than slower NICs). This is one of the reasons why connections can easily get dropped when transitioning from an area with low-speed wireless coverage to higher speed wireless coverage, since the routing table manager blindly switches connections even if the other connection is still available.

As the figure shows, the MultiAdapter inserts itself as a virtual miniport adapter (from the point of view of the actual protocol adapters) and as a set of virtual protocol adapters (from the point of view of the actual miniport adapters). The protocol adapters now only see one underlying miniport driver, and therefore will always use the MultiAdapter's virtual miniport to transfer its data. Internally, the MultiAdapter establishes a binding to the TCP/IP protocol adapter (referred to as the miniport upper edge) and to each of the underlying miniport adapters (referred to as the protocol lower edge) using the interface exported by the NDIS (Network Driver Interface Specification) library. During this initialization phase, the MultiAdapter also assigns a MAC address to the virtual miniport (we simply use the same MAC address as one of the NICs). Once the bindings are established, the MultiAdapter determines the status of each of the underlying miniports, and forwards the data based on the following policy.

The MultiAdapter distinguishes the traffic flowing through it into two main categories. One is http traffic, identified by the application header, which gets routed through the fastest adapter that is connected and available. The other consists of all remaining traffic, which is channeled through the GPRS interface whenever it is available, and when it is not, through the Wi-Fi interface. This results in the behavior such that all applications other than the web browser are provided with the more persistent connection albeit the slower transfer speed, while the web browser, owing to its stateless nature and richer content, is given access to the fastest connection available. Each time the connection status of the underlying miniport adapters change, the MultiAdapter receives a notification through the NDIS interface and appropriately updates the list of virtual adapters through which it can route traffic.

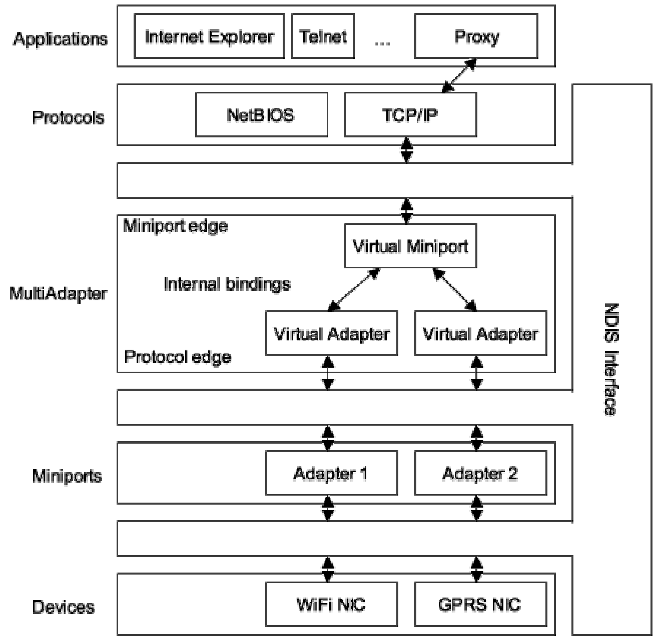

Figure 1: System Architecture Diagram

We built the Caching System and local HTTP proxy in C# on the .NET Framework 1.1 for the Windows OS. This system consists of approximately 3,000 lines of code. As stated earlier we configure web browsers to use our local HTTP proxy as their method of accessing the Internet and when connected a copy of each outgoing HTTP header is sent to the Caching System.

When an HTTP header arrives in the Caching System it is put in the Caching Classifier's queue. Each of these headers is then classified into one of five groups (as described in the system design section). It is then added to the caching queue.

The caching system monitors the caching queue and when there is an available network connection of sufficient bandwidth, it caches the resource. This is done in a manner such that items in higher priority classifications get cached sooner than those of lower priority. Additionally, when a page is cached all child pages not already in they queue are cached as well (click-ahead caching).

Over time, items in the cache may become outdated. To deal with this the caching system watches for aging content according to the last time it was updated and its caching classification. High priority items that were cached the longest ago get added to the cache queue first. Then other items are added to the queue as they become old enough. Thus, if a user accesses say a news website then has intermittent Wi-Fi access throughout the rest of the day, it is likely the news website will, in fact, get updated in the cache with the new news so that it may be accessed offline.

PRELIMINARY EVALUATION

We compare the user perceived performance of the three representative applications, based on the number of connection interruptions and failed network request attempts, under our user scenario without our system and with our system. We used a laptop computer operating Windows XP with built-in Wi-Fi transceiver and a GPRS PC card, running Internet Explorer for the web application, Yahoo Messenger for the IM application, and SSH for the remote terminal application.

In this preliminary evaluation we evaluate the caching system. We compare the caching classifier system with a control system in a field test. Note that the operating system we use, MS Windows XP, will automatically down-switch from Wi-Fi to GPRS when it loses the Wi-Fi connection, and reconnect to Wi-Fi when it becomes available.

The computer remains open for the extent of the evaluation. While less reasonable on a laptop, some people carry laptops this way and it is analogous to the behavior of a PDA or other portable device. We study the user's connectivity experience in terms of a realistic scenario with representative connectivity. We then observe the user's experience and connectivity characteristics of the protocols.

Evaluation of the Caching System

a) The user begins in the Allen Center on the main floor. The Allen Center has a Wi-Fi network open to the user and there is occasional GPRS coverage. She browses the web, going to http://www.chi2005.org/, a conference homepage. She also visits news.google.com, and clicks on the top three articles, the top US article, and the top sports article. Next, she checks her email using Pine over SSH, for approximately 5 minutes. Then she has a brief IM conversation with a colleague.

In both the control and the caching condition, the Wi-Fi was available in the Allen Center, but GPRS was unavailable or spotty. The performance of the web page responses was the same in both conditions, from immediate to a couple of seconds. Note that in the caching condition, while the user is checking her email, the caching system is using the leftover bandwidth to proactively refresh the web pages she has visited and click-ahead cache the links on those pages.

b) She then walks to the main floor of Engineering Library among the stacks. The Engineering library does not have an accessible access point for a Wi-Fi network, and GPRS is not accessible. The first task at this location is to go to news.google.com, look at the top story, and two other articles not read earlier. A second task is to go back to the conference web page, http://www.chi2005.org/, and find out more information about the late breaking abstracts and the registration deadlines (available on secondary pages).

This area has no connectivity, so in the control condition, there was no response. The Google News page and the conference page had been cached by Internet Explorer, but the Google news page was out of date and the user could not access at all any of the secondary pages on either Google News or CHI2005. In the caching condition, the responses were immediate, and the caching system had cached all the Google News stories accessed, including the advertisements. For the CHI2005 secondary pages, it had cached the information on the pages, but had not prioritized or cached the photographs of the late breaking results chairs on one page.

c) The next location is upstairs in the Engineering Library. Here, there is GPRS connectivity, but still no Wi-Fi. The task is to go to www.google.com, do a search, and begin composing an email. The GPRS took an about 13 seconds to return the google.com main page (of size 2.51KB). It took 10s to return the search results, while the server response time was a negligible 0.16s. In the control case, the response times were much faster (about 1~2 s). We attribute this to the variability of GPRS, and not an effect of our system. The longest time needed to login to SSH and open the inbox of Pine on GPRS was 27s, not counting the user's typing time. The user leaves the email open and proofreads it as he walks back to the Allen Center.

d) As the user returned to the Allen Center, in both cases the GPRS connection was dropped when the Wi-Fi came in range. This caused the SSH to drop the connection and refuse to accept user input, even though the GPRS connection it had been using a moment before was still at full connectivity. This is the challenge that the Multi-Adapter was designed to correct, and it will be discussed in more detail in the next section.

DISCUSSION

The results of our preliminary evaluation point to the advantage the caching system has over the control condition. In areas of no connectivity, the caching system immediately returns updated results on pages that are linked to those used in the past. Without this system, users can only look at outdated versions of pages they have seen before.

The evaluation also demonstrated in both conditions the need for our MultiAdapter's functionality. When the Wi-Fi became available again, the SSH connection was dropped and the email was lost despite the GPRS still being available. We believe the MultiAdapter would have acted in a way more consistent with the user's needs by continuing to use GPRS for the SSH connection as long as the GPRS connection was available while letting other applications like the web browser use the newly acquired Wi-Fi connection.

While our approach to caching is similar to previous work in web caching, the main contribution of this work is the combination of our caching approach and the simultaneous multiplexing of multiple wireless technologies. This not only allows our caching system to be more aggressive with respect to refreshing frequently changing and requested content, it allows the system to provide differing types of service (if available) to different applications simultaneously.

Future advances in wireless technologies promise greater coverage at higher speeds. However, new wireless technologies do not completely alleviate the problems our system addresses. For example, all currently proposed technologies will still have many coverage gaps such as a parking garage, the basement of an old building, or a very rural road. In these cases the ability to provide the best offline content is useful. Additionally, it is still likely that different wireless technologies will continue to offer differing service with differing coverage making the idea of simultaneous multiplexing of wireless technologies still relevant.

FUTURE WORK

The current status of our system implementation is a working Caching System and a partially functioning MultiAdapter. Our most immediate future work is to complete implementation of the MultiAdapter and evaluate the system as a whole in a field trial.

There are also other areas that we would like to explore once the system is fully implemented. One is in the area of intuitive user interface for both notification of changing network status and user specification of preferences. As the system automatically determines when it is most appropriate to draw content from the cache and which connection it should use, the question arises concerning what is the best way to portray to the user which mode the system is operating under. For example, if the system decides to pull a web page from the cache instead of the Internet, it may be crucial that the user is made aware of the fact that the content she is looking at is not live. Also, in order for the user to gain control over the caching criteria or in specifying the connectivity versus bandwidth characterization of certain applications, there must be an intuitive and simple interface through which the user can specify the appropriate choices. Our caching system has the potential to improve performance in areas of slow (GPRS) connectivity as well as areas of no connectivity, and more work needs to be done looking at cache content freshness and when the cached data can be returned instead of waiting for the latest version over a slower connection.

Another area that we would like to look at is in enhancing the connectivity to available Wi-Fi networks, such that it can seamlessly connect to open hotspots or those on which the user has an account. Management GUI for such information also becomes an issue. We would also like to look into combining Mobile IP with our system such that we enable seamless connectivity across multiple networks as well as across heterogeneous connection types. This would increase the extent of high-bandwidth regions that are available to the user.

We would also like to explore our caching metrics in more detail, evaluating the efficacy of the heuristics we have presented and looking at statistical long-term measures of prediction versus hit rate and optimization of bandwidth usage.

CONCLUSION

We have presented a two-pronged approach to solving the problem of continuous connectivity. We present a proactive caching system to store and update likely candidates for web pages to be viewed in the future. We also discuss the MultiAdapter, a new approach that uses multiple channels to support the most important connection needs of various applications. Our system was evaluated in an example scenario and increased user perception of connectivity, allowing access to web pages that would have been otherwise unavailable.

REFERENCES

1. An Architecture for Integrating UMTS and 802.11 WLAN Networks Muhammad Jaseemuddin. http://www.ee.ryerson.ca/~jaseem/pub/UMTS_80211_iscc03.pdf.
2. BARWAN: Bay Area Research Wireless Access Network. Katz et al. COMPCON 1996.
3. Mobile computing with the Rover toolkit. Anthony D. Joseph, Joshua A. Tauber, and M. Frans Kaashoek. IEEE Transactions On Computers, Vol 46 Nov 3 March 1997.
4. IP Routing for Wireless/Mobile Hosts (mobileip). http://www.ietf.org/html.charters/mobileip-charter.html
5. WiMAX: Broadband Access Technology. http://www.intel.com/netcomms/technologies/wimax/
6. Integrating Wi-Fi with Cellular http://www.dailywireless.org/modules.php?name=News&file=article&sid=225
7. AvantGo http://www.avantgo.com/frontdoor/index.html
8. PAWNs: Satisfying the Need for Ubiqutious Secure Connectivity and Location Services. Bahl et al.
9. "A Non-interfering Deployable Web Prefetching System," R. Kokku, P. Yalagandula, A. Venkatramani, M. Dahlin,, Proceedings of the USENIX Symposium on Internet Technologies and Systems, Mar 2003.
10. "Potential costs and benefits of long-term prefetching for content-distribution," A. Venkataramani, P. Yalagandula, R. Kokku, S. Sharif, M. Dahlin, Elsevier Computer Communications, 25 (4) Mar 2002.
11. Wenzheng Gu and Abdelsalam (Sumi) Helal, "Extended Internet Caching Protocol: A Foundation for Building Ubiquitous Web Caching," Proceedings of the ACM Symposium on Applied Computing (SAC), Melbourne, Florida, March 2003.
12. H. Kobayashi and S-Z Yu, "Performance Models of Web Caching and Prefetching for Wireless Internet Access" International Conference on Performance Evaluation: Theory, Techniques and Applications (PerETTA 2000), September 21-22, 2000, University of Aizu, Fukushima, Japan.

13. Wu,, T., S. Ahuja and S. Dixit, "Efficient Mobile Content Delivery by Exploiting User Interest Correlation", WWW 2003, Budapest, Hungary, May 2003.

14. Data Capabilities: GRPS to HSDPA. http://www.3gamericas.org/pdfs/rysavy_data_sept2004.pdf

15. The IEEE 802.16 Working Group on Broadband Wireless Access Standards. http://wirelessman.org/

16. Intel Network Connectivity – Adaptive Load Balancing. http://www.intel.com/network/connectivity/resources/technologies/load_balancing.htm

17. C. Perkins, "Mobile IP," IEEE Commun. Mag., vol. 35, pp.84–99, May 1997.

18. James J. Kistler , M. Satyanarayanan, Disconnected operation in the Coda File System, ACM Transactions on Computer Systems (TOCS), v.10 n.1, p.3-25, Feb. 1992