Mobile Computing
- DataMan Project Perspective

Tomasz Imielinski
Rutgers University

1 Introduction

The objective of mobile computing is to develop system and application level software for small, battery powered terminals equipped with the wireless network connection. There is a rapidly growing interest in this field with companies spending billions of dollars in the recent PCS a ctions showing high expectations of a future massive market for wireless information services for mobile users. The objective of this paper is to provide a perspective on research issues in mobile computing from the standpoint of our own research project. The *DataMan*[1] laboratory involves four major research projects:*wireless web*, *infostations*, *geographic messaging* and *satellite information systems*. We describe this project in more detail in the next section. Before we do that we will briefly summarize the key research challenges brought by the mobile and wireless environment.

1.1 Research Challenges

Here we provide only a brief overview of the new research issues arising in mobile computing. For more elaborate discussion the reader is referred to [16], [14], [15].

We will assume here the basic architecture of the mobile environment which consists of MSS (Mobile Support Stations) - base stations with limited radio coverage. The area covered by a base station is called a *cell* and its size varies from several miles (cellular AMPS system) to perhaps tens of meters in case of infrared picocells and wireless LAN. The future PCN (Personal Communication Network) assumes a wide variety of cells starting from very small picocells coexisting with larger, macrocells which may even include satellite radio [25].

We argue in [13] that the following features of the mobile environment will have a critical impact on the software design issues for mobile computing:

- Massively Distributed Architecture

  The concept of a cellular architecture aimed at efficient frequency reuse implies a massively distributed system architecture. Thus, better bandwidth utilization indirectly leads to increased system complexity. The resulting system issues include handoff between cells on all OSI layers, as replication, migration and caching of data related to user transactions as well as user profiles. Consequently, both server and client software has to be designed to incorporate adaptive replication and migration as well as handoff. Several researchers proposed

[1]DataMan Project is funded by DARPA and NSF grants (ARPA under contract number DAAH04-95-1-0596, NSF grant numbers CCR 95-09620, IRIS 95-09816) with T. Imielinski as the Project Director and B.R. Badrinath as the coDirector

new models such as client-proxy-server to handle interactions between the clients and servers more smoothly, especially when clients are moving.

- Client and Server Mobility

  Mobility affects network layer issues such as routing, and is currently addressed comprehensively in the MobileIP proposals [26]. Additionally, location is an important parameter in location dependent applications. Finally, location and mobility play a central role in geographic messaging [10], which we discuss later in this paper.

- Widely varying communication conditions: Fading, Noise and Interference

  This requires adaptive solutions on all layers of the OSI hierarchy, handling failures and disconnections. Here again, the concept of a proxy will be useful to reduce the fundamental mismatch between the wireless "last mile" and the fixed network. For example a number of proposals to extend TCP to better handle wireless environments are based on the concept of a TCP proxy which resides on the MSS [16]. Other important issue with far reaching impact on file systems and databases is *disconnection*, Disconnection can be viewed as prepared or voluntary failure. CODA, Locus and other distributed file systems were the first to deal with the disconnection on the file system level. In databases disconnection is handled typically in the context of long term transactions.

- Power Considerations

  Energy is an important constraint due to the battery life limitations. As pointed out by [6], the battery life is not expected to increase more than 20 percent in the next 10 years. Hardware manufacturers are providing hardware solutions such as spinning down the disk and switching off the screen. It is necessary now to provide energy efficient software solutions. Some initial work in this direction is provided by Weiser (included in [16]).

- Varying Tariff Structures

  The future tariff structure for wireless information services is far ¿from clear yet. Basically, charges can be proportional to the connection time, number of packets sent or received, or simply could be flat up to a certain limit. Additionally, charges may depend on the time of the day and even on the distance from the base station. The type of tariff will have strong impact on the software running on he client machine - different solutions will be used for different types of tariffs. For example [14] prefetching will be used aggressively, when charges are based on connection time, while it will be avoided when charges are based on the number of packets exchanged.

Our own work within DataMan project deals with most of the issues described above. Below, we provide a more detailed description of the key projects.

2 Mobile Computing at Rutgers

In this section we provide an overview of some of the major projects which we are involved at Mobile Computing Laboratory at Rutgers University. The work is organized into four major projects: *infostations*, *geographic messaging*, *wireless web* and *satellite information systems*. Infostation project is an interdisciplinary effort between the DataMan Lab and WINLAB, with R.Frenkiel, D.Goodman, J.Evans and T.Imielinski as principal investigators. The project involves building radio and protocol support for *infostations* - wireless information kiosks with high bit rate and limited range. It addresses a broad range of issues starting from the low level radio design and ending on application protocol support. The geographic messaging project deals with a hypothetical scenario in not so remote future when each terminal carries a GPS card. We are building routing support for delivering messages to the specific geographic destinations specified as polygons of GPS coordinates. We are also providing support for client/server applications where location and distance play a role is server's selection. In wireless web, we provide new implementations for the existing applications such as ftp, e-mail and even http which are better suited for the wireless medium. Finally, the project on satellite information systems deals with the design of a system with highly asymmetric bandwidth which uses high bit rate donwlink channel to periodically broadcast possibly large data items. The uplink channel is used primarily for queries.

Below, we discuss each of these projects in more detail:

2.1 Geographic messaging and Geographic Information Services

In the near future GPS will be widely used allowing a broad variety of location dependent services such as navigation, traffic information etc. In this project we propose a family of protocols and addressing methods to integrate GPS into the Internet Protocol to enable the creation of location dependent services such as:

- Multicasting selectively only to specific geographical regions defined by latitude and longitude. For example, sending an emergency message to everyone who is currently in a specific area, such as a building or a train station.
- Providing a given service only to clients who are within a certain geographic range from the server (which may be mobile itself), say within 2 miles.
- Advertising a given service in a range restricted way, say, within 2 miles from the server,
- Providing contiguous information services for mobile users when information depends on the user's location. In particular, providing *location dependent bookmarks*, which return web pages describing services located within a certain distance from the user.

Below, we list some possible scenarios of usage for the geographic messaging.

Consider an example situation, of an area of land near a river. During a severe rain storm, the local authorities may wish to send a flood warning to all people living within a hundred meters of the river.

For the interface to such messaging system we propose to use a zoom-able map similar to the U.S. Census Bureau's Tiger Map Service. This map would allow a user to view a geographical area at varying degrees of magnitude. He could then use a pointing device, such as a mouse, to draw a bounding polygon around the area which will receive the message to be sent. The computer would then translate the drawn polygon into GPS coordinates and use those coordinates when sending and routing the message. Geographical regions specified using this zoom-able map could be stored and recalled at a later time. This zoom-able map is analogous to the IP address books found in many email programs.

To continue with the above example, local officials would call up a map containing the river in danger of overflowing. They would then hand-draw a bounding polygon around all of the areas at least a hundred yards from the river. They would specify this to be the destination for a flood warning email to all residents in the area. The warning email would then be sent. Similar applications include traffic management (for example, reaching vehicles which are stuck in traffic) and security enforcement.

The main challenge is to integrate the concept of physical location into the current design of Internet which relies on logical addressing. We study the following general families of solutions:

- Multicast solution
- Unicast IP routing extended to deal with GPS addresses
- Application Layer Solution using extended DNS

Here we discuss only the first solution - the complete discussion is provided in [10]

We will assume a general cellular architecture with base stations called Mobile Support Stations (MSS). We will consider a wide variety of cells, including outdoor and indoor cells. We will discuss both cases when the mobile client has a GPS card on his machine and cases when the GPS card does not work (i.e.inside buildings).

2.1.1 Addressing Model

Two-dimensional GPS positioning offers latitude and longitude information as a four dimensional vector:

¡Direction, hours, minutes, hundreds of minutes¿, where Direction is one of the four basic values: N, S, W, E; hours ranges from 0 to 180 (for latitude) and 0 to 90 for longitude, and finally, minutes and hundreds of minutes range from 0 to 60.

Thus ¡W, 122, 56, 89¿ is an example of longitude and ¡N, 85, 66, 43¿ is an example of latitude.

4 bytes of addressing space (one byte for each of the four dimensions) are necessary to store latitude and 4 bytes are also sufficient to store longitude. Thus 8 bytes total are necessary to

address the whole surface of earth with precision down to 0.1 mile! Notice that if wanted precision down to 0.001 mile (1.8 meters) then we would need just 5 bytes for each component, 10 bytes together for the full address (as military versions provide). The future version of IP (IP v6) will certainly have sufficient number of bits in its addressing space to provide a multicast address for even smaller GPS addressable units. In [10] however, we assume the current version of IP (IP v4) and we make sure that we manage the addressing space more economically than that.

By GPS square we mean the smallest GPS addressable unit.

A destination GPS address would be represented by one of the following:

- point(GPS-square address)

  This notation would send a message to anyone within a specific GPS-square.

- Some closed polygon such as: circle(center point, radius) polygon(point1, point2, point3, ... , pointn, point1)

  where each point would be expressed using GPS-square addresses. This notation would send a message to anyone within the specified geographical area defined by the closed polygon.

- site-name as a geographic access path

  This notation would simulate the postal mail service. In this manner, a message can be sent to a specific site by specifying the its location in terms of real-world names, by specifying a sequence starting from the specific site, city, township, county, state etc. This format would make use of the directory service detailed later.

For example, if we were to send a message to city hall in Fresno, California, we could send it by specifying either a closed polygon or the mail address. If we specify a bounding polygon, then we could specify the GPS limits of the city hall as series of connected lines that form an enclosed polygon surrounding it. Since we have a list of connected lines, we just have to record the endpoints of the lines. Therefore the address of the city hall in Fresno could look like:

polygon([N 45 58 23, W 34 56 12], [N 23 45 56, W 12 23 34], ...)

Alternatively, since city hall in Fresno is a well-defined geographical area, it would be simpler to merely name the destination. This would be done by specifying "postal-like" address such as $city_hall.Fresno.California.USA$.

For "ad hoc" specified areas such as, say a quad between 5th and 6th Avenue and 43 and 46 street in New York, the polygon addressing will be used.

Unfortunately, we will not be able to assume that we have enough addressing space available in the IP packet addressing space to address all GPS squares. Instead we will propose a solution which is flexible in terms of the smallest GPS addressable units which we call atoms. In our solution, a smaller available addressing space (in the IP packet) will translate into bigger atoms. Obviously, we can use as precise addressing as we want to in the body of the geographic messages - the space limitations apply only to the IP addressing space.

By a geographic address we mean an IP address assigned to a geographic area or point of interest. Our solution will be flexible in terms of the geographic addressing space.

Below, we will use the following two terms:

- Atoms: for smallest geographic areas which have geographic address.

  Thus, atoms could be as small as GPS squares but could be larger

- Partitions: These are larger, geographical areas, which will also have a geographic address. A state, county, town etc. may constitute a partition. A partition will contain a number of atoms.

Here are some examples of possible atoms and partitions:

- A rectangle, defined by truncating either longitude or latitude part of the GPS address by skipping one or more least significant digits

- A circle, centered in a specific GPS address with a prespecified radius.

- Irregular shapes such as administrative domains: states, counties, townships, boroughs, cities etc

Partitions and Atoms (which are of course special atomic partitions) will therefore have geographic addresses which will be used by routers. Areas of smaller size than atoms, or of "irregular shape" will not have corresponding geographic addresses and will have to handled with the help of application layer.

2.1.2 Routing

Let us now describe the suggested routing schemes responsible for delivering a message to any geographical destination. The discussion will begin by describing the infrastructure assumed by the routing schemes. This infrastructure assumes that both mobile computers and fixed-network computers will be the targets of GPS-addressed messages.

2.1.3 GPS-Multicast Routing Scheme

Here, we discuss the first leg of routing: from the sender to the MSS. Each partition and atom is mapped to a multicast address. The exact form of this mapping is discussed further in this subsection. We first sketch the whole concept.

This solution provides flexible mix of the multicast and application level filtering for the geographic addressing. The key idea here is to approximate the addressing polygon of the smallest partition which contains it and using the multicast address corresponding to that partition as the IP address of that message. The original polygon is a part of the packet's body and the exact matching is done on the application layer in the second leg of the route.

The basic idea for the first level of routing using multicast is to have each base station join multicast groups for all partitions which intersect its range. If the proper multicast trees are constructed than the sender can simply determine the multicast address of the partition which covers the original polygon he wants to send his message to, use this multicast address as the address on the packet and put the original polygon specification into the packet content. In this way, multicast will assure that the packet will be delivered to the proper MSS.

Example

For instance the MSS in New Brunswick may have its range intersect the following atoms and partitions: Busch, College Avenue, Douglass and Livingtson Campuses of Rutgers University (atoms), New Brunswick downtown area (atom), the Middlesex county partition and the NJ state partition. Each of these atoms and partitions will be mapped into a multicast address and the New Brunswick's MSS will have to join all such multicast groups.

However, things cannot be as simple as described. For such a large potential number of multicast groups if we build entire multicast trees, the routing tables could be too large. Fortunately it is not necessary to build complete multicast trees. Indeed, it in not important to know precise location of each atoms in California, from the location in NJ.

Thus, we modify our simple solution by implementing the following intuition:

The smaller is the size of the partition (atom) the more locally is the information about that partition (atom) propagated.

Thus, only multicast group membership for very large partitions will be propagated across the whole country.

For example, a base station in Menlo Park, California can intersect several atoms) and several larger which cover Menlo Park, such say a parttion which covers the entire San Mateo county, next which cover the entire California and finally next which may cover the entire west coast. This base station will have to join multicast groups which correspond to all these rectangles. However, only the information about multicast group corresponding to the West Coast partition will be propagated to the East Coast routers.

However, a simple address aggregation scheme in which only a "more significant portion" of address propagates far away would not work. Indeed, in this case a remote router, say in NJ, could have several aggregate links leading to California - in fact, in the worst case, all its links could point to California since it could have received a routing information to some location in California on any of those links.

To avoid this, for each partition we distinguish one or a few MSS which act as designated router(s) for that partition. For example, the California partition, may have only three designated routers, one in Eureka, another in Sacramento and yet another in LA. Only the routing entries from the designated routers would be aggregated into the aggregate address for California. Information coming from other city routers will simply be dropped and not aggregated at all. This, in addition to a standard selection of the shorter routes, would restrict the number of links which lead to an aggregate address. In particular, when there is only one designated router per partition, there would only be one aggregate link in any router. This could lead to nonoptimal routing but will solve the problem of redundant links [10].

We still have to describe what happens when a message reaches the MSS. There is a number of alternatives and again we only discuss one of them here, The MSS compares the polygon coordinates in the message with the range of its own cell. If there is no intersection than the message is dropped by MSS. If there is an intersection than MSS multicasts a short message on a well known multicast address which asks those clients whose current location falls into the intersection to join a temporary multicast group. Then the full message is multicasted by the MSS on that multicast address.

The detailed nature of the mapping from atoms and partitions to multicast addresses is described in [10].

In [10] we also discuss two other methods of delivering geographic messaging: geometric routing and domain server solution.

In the domain server solution the geographic information is added to the DNS which provides the full directory information down to the level of the IP address of each base station and its area of coverage represented as a polygon of coordinates.

A new first level domain - "geographic" is added to the set of first level domains. The second level domain names include states, the third, counties and finally, the fourth: polygons of coordinates, or so called points of interests. We can also allow, polygons to occur as elements of second, third domains to enable sending messages to larger areas.

Thus a typical geographic address can look like

city-hall-Palo-Alto.San-Mateo-County.California.geographic

or

Polygon.San-Mateo-County.California.geographic

where Polygon is a sequence of coordinates.

This geographic address is resolved in a similar way as the standard domain addresses are resolved today into a set of IP addresses of base stations which cover that geographic area.

A prototype implementation is carried through at the Rutgers Campus.

2.1.4 Distance Based Services

The location awareness enabled by GPS allows support for distance based services which are located within a certain distance ¿from the mobile client. In [10] we propose the following simple solution allowing:

- Servers to advertise their services within a certain distance from their current location[2]
- Clients to request services only within the certain distance ¿from their current location.

We assume again that both clients and servers are equipped with the GPS cards.

We first describe a static solution when neither clients nor server move.

[2]Servers may be mobile as well

A pair of multicast addresses is used to define the two versions of the same service. Mulitcast address S is used for the server to advertise its service to the clients. This mode refers to the *server initiated* service delivery. Multicast address C is used for the clients to inquire about the same service. This is the *client initiated* service delivery mode and C is used for *service queries.*

Thus, clients multicast their service requests on the address C and servers multicast their advertisements on S. Thus, a client who wants to receive specific service advertisements has to join the S group and the server which wants to respond to the client's queries joins the multicast group C. For example, suppose we deal with the traffic service. The traffic server may periodically multicast traffic information on the address S and clients who want to receive the traffic data will join that group. Similarilly, servers which want to respond to client requests will join another traffic group C.

Now, let us discuss how to generalize this scheme to a situation when both servers and clients may be mobile. In this case we propose that both S and C addresses are concatenation of two parts: the service name and the location of the server (client) expressed as an atom of our addressing scheme. Thus the final addresses on which clients multicast their queries and servers advertise their services will depend on the client's (server's) location. For example, the server located in New Brunswick which wants to advertise traffic information only to the New Brunswick atom (which may be city limits, for example) will use a concatenation of the service identifier and the bit string corresponding to the New Brunswick atom. Thus, traffic service in Princeton will use different "location part" in its S address. Clients who want to listen only to "local ads" will then join the S address which corresponds to the atoms they currently reside in. In this way a client who is currently located within the city bounds of New Brunswick will only listen to the ads within the New Brunswick atom. Similarilly, for the client initiated mode, the client requests will only go to the "near by" servers which again will join only C addresses for the "near by" clients. This schemes generalize the earlier anycast, narrowcast and nearcast proposals to the situation when the GPS cards are used. The presence of GPS card is necessary both for the clients as well as servers in order for them to determine the current S and C addresses to join.

We are currently developing a prototype of the extension of the WWW browser in which the user will be able to specify graphically on the local map restrictions on the locations of Web pages which are of interest. Thus, for example, a user may specify that he may only want to see web pages located on a server within a given building.

In general, the geographic messaging project introduces location as a "first class citizen" both in message addressing as well as in service discovery. The next project shows that location may also be critical from the quality of service standpoint.

2.2 Wireless Web

TCP/IP protocol suite which is the foundation of Internet has not been designed with the wireless medium in mind. TCP which is a stepping stone for such popular applications as ftp and http is not well suited for wireless link. Additionally, application protocols such as ftp, http and the family of e-mail protocols have not been created with wireless links in mind. Thus, they typically lack features to handle frequent disconnections, handoffs, connection failures and varying bandwidth conditions. The main objective of the wireless web project is to provide suitable extensions of the existing application and transport protocols to handle wireless links better.

2.2.1 Wireless and Asymmetric TCP

TCP interprets packet loss as a result of congestion, while in fact, on the wireless link the packet loss in mainly due to the higher error rate. Thus, TCP instead of "trying harder" when facing the packet loss, backs off which is a correct behavior in case of congestion but degrades throughput on the wireless link. Bakre and Badrinath [23], [22], [5] implemented I-TCP, indirect TCP which splits the TCP connection into two pieces: one on the fixed network between the sender and the MSS and another on the wireless link between the MSS and the moble client. Only the second leg of the connection between the MSS and the mobile client needs to be modified while the connection between the sender and the MSS is handled as usual. This work (which is Bakre's PhD thesis) has led to the implementation prototype and is now extended further. Additionally, TCP has been designed assuming symmetric communication link between the sender and the receiver. In many situations in wireless environments the links are in fact highly asymmetric. For example in satellite environment, downlink channel may carry bit rates of the order of Mb/s (Direct PC), while uplink channel is only capable of carrying perhaps only several kb/s. This requires a careful approach in sending acknowledgments from the receiver to the sender. Additionally, since energy required to transmit is much higher than the energy consumed by the receiving process, we want to avoid sending acknowledgments too often possibly draining the limited battery resources on the client's end.

We are currently examining combinations of extensions of UDP (possibly with negative acks) o the wireless link in conjunction with the TCP running between the sender and the client.

2.2.2 Opportunistic End to End Protocols

The mobile computing environment is associated with changing resource constraints. Protocols and applications must react to changes so that protocols perform optimally under the current environment and available resources are properly utilized. As mobile hosts move, they encounter changing network characteristics. Characteristics such as bandwidth, connectivity, reliability and cost can change drastically when a mobile host moves.

Existing end-to-end protocols are designed with an assumption that network conditions don't change drastically. In mobile environments, protocols must be made aware of underlying network conditions so that they can react to the changes to perform optimally. For example, due to mobility of hosts, there can be transient yet significant packet loss when the mobile is far away ¿from a base station. Unreliable protocols such as UDP will perform poorly if packets are transmitted when signal quality is bad. A mechanism by which unnecessary loss of packets can be avoided by allowing transmission only under "good" link conditions is presented in [2].

In general, end-to-end protocols need information about the state of the wireless link. The decision of what to do under a given condition can be taken at various layers such as device level, IP level and transport level. At each level, the information sought about the conditions and the action taken will vary. For example, if signal quality is bad at the transport level, a feedback of network outage can be used to freeze the retransmission timers. What level the decision should be made and what information should be used by the protocol raises a number of research issues about layering and protocol design. Protocols that make use of state information provided by lower layers need a concept of "punch through layering" which is currently being developed.

2.2.3 Opportunistic application protocols

We are working on extensions of ftp and e-mail protocols allowing the user to specify conditions under which ftp session can occur, or when sending/receiving of e-mail may take place. Under opportunistic scenario the ftp commands are not necessarily executed immediately, but are deferred until appropriate (prespecified) conditions occur. Such condition typically involve the quality of the channel (bit rate, error rate, cost per packet) but may also involve other parameters such as power level on the client's machine.

Consider for example a scenario when the user's machine is equipped with two wireless interfaces: *CDPD* and *Wavelan*. A user may specify that in order to download a large file the client must have wavlan connectivity. Thus, if only *CDPD* connection is available the ftp request will be deferred and queued until the wavalan connection is available. In our current implementation the priority list will be specified in the form of ordered list of interfaces. Thus, for instance, if *Wavelan* interface is ordered "above" *CDPD* interface than all operations allowed under *Wavelan* interface will be also allowed under the *CDPD* interface (not vice versa, however). In general, all operations allowed under the interface "i" will be allowed for all interfaces ordered "after" i, thus for k¿=i.

Additionally, in the currently developed implementation,the atomicty of an ftp session will not be restricted to one interface and one TCP connection. Thus, the state of an ftp session will be carried through different connections on different interfaces and possibly even from different locations. For example, a user who is in the middle of downloading a large file through *CDPD* may discover the availability of *Wavelan* connection. This may cause termination of the *CDPD* connection and opening the new connection via *Wavelan* link and continuing the file operation from the position where it was interrupted.

Similar features are implemented as a part of the wireless e-mail implementation. Here, the user may specify conditions under which he will download e-mail on the mobile computer through a wireless links. Additionally, he can place different filters which will allow only certain important pieces of e-mail to be downloaded to the clients machine.

The opportunistic nature of transmission is discussed in [2] and the general model of communication under varying conditions in [4].

Mobile applications need support for obtaining the state of the current environment and any changes that may occur as the mobile host moves. As resource availability changes so should the applications need for resources. In order to effectively adapt to changes in resources, applications should contribute by controlling resource allocation and not leave the decision solely to the operating system. For effective application adaptation, awareness of the environment need to be exported to the application. A software architecture that includes abstractions to communicate mobility related events at the language level is described in [4].

Application software designed for mobile devices has to incorporate mechanism to handle changes in environment when such changes are reported. The basic mechanism that is currently being developed is based on the concept of an event channel. Both system level events and application level events are reported to an event channel. Applications provide handlers that subscribe to this event channel. A comprehensive event delivery mechanism and policy is used to deliver events to the handlers. Currently, we are building applications that can adapt to changes in the mobile environment using the concept of an event channel.

We are also investigating efficient extension of web protocols such as http for wireless links. The key idea in this implementations is the concept of a proxy.

2.2.4 Proxies and http

To avoid separate TCP connections for every page accessed on the Web from the wireless client we modify HTTP in such a way that a proxy will be opening and closing connections and the client/proxy connection is managed separately. for example through a simpler reliable protocol (RDP). An implementation of RPC over RDP/UDP exists. Using this as a starting point we design an indirect http protocol which uses rdp/tcp combination. Other features include grouping several connections, prefetching pages that refer to files at the same server (when a request for a page is made, prefetch hyperlinks on this page that are on the same machine), text only pages and any other techniques that can reduce latency of displaying a page on a mobile client connected over a low bandwidth wireless link. The HTTP proxy will also perform other functions dealing with changing the format of the page to better fit to the wireless medium (for example no images etc).

Also, one could add to the HTML authoring tool special new annotations which would display a given element of the page only in certain resource prerequisites are satisfied.

Thus, for example an image will appear only if there is enough bandwidth etc.

Important issue is whether proxies will move along with users or whether they will stay in the same location. For example, using the mobile IP terminology, the proxy (containing the profile of the user) may reside at a home agent or at the foreign agent [26]. Proxy at home agent will not move but it would require the home agent to be a part of connection between the sender and the receiver, which we would like to avoid, except for the few initial packets. Proxies residing at foreign agents require careful mobility management especially for users who change locations very often.

The preliminary work using the applications of the proxy model to the remote procedure call (rpc) is discussed in [3].

2.3 Infostations

The objective of this project[20] is to develop architecture, software, protocol, and radio support for infostations - pockets of high bit rate wireless connectivity [20]. Infostations will serve as information kiosks for mobile users equipped with terminals with wireless connection. Key new ideas are that spectrum efficiency and high bandwidth can be achieved if range is limited, and that innovative radio technology can allow access at very low cost. This leads to an architecture involving "islands of coverage", at which messaging services, web access and innovative new services can be offered. Users can communicate as they pass an infostation, or remain in the area during an ongoing "session". Technical issues involve radio design, software and protocol design as well as general system architecture. The numerous commercial applications include enhanced messaging and web services for mobile users, and innovative new services for shopping, traveling, finance, security and equipment or inventory management.

The coverage area of infostations is small and discontinuous (i.e., infostations are separated from one another by many times their radius of coverage). This allows all the available spectrum to be used at every infostation. Moreover, as the user moves closer to the infostation, the quality of the channel is improved. The resulting bandwidth and channel quality allow the infostation to carry very large data rates efficiently and economically. Since the coverage is discontinuous, the moving user will only have access to an infostation periodically, which makes the architecture particularly suitable for information which can be delayed. Many telecommunications applications including e-mail, fax and voice main tolerate delays of a few minutes if a user is willing to wait, s/he will be able to utilize a cheaper and faster service than can be provided by one, such as cellular or PCS, with ubiquitous coverage. Thus, infostations offer a tradeoff resulting in much higher effective bitrate and lesser cost in return for tolerable delay. In simple terms - "if you are willing to wait for a few minutes you can spend a few cents rather than a few dollars on downloading a wireless fax to your machine".

Infostations can be coordinated with coverage provided by systems such as cellular, PCS, satellite and paging systems. These systems can provide an underlying and ubiquitous "backbone" coverage which can transfer information at relatively low bit rates, but which is expensive and slow for the transfer of large files. Used in combination with infostations, these systems allow the user to

Because of the numerous applications which are possible, a significant activity will involve the identification of services and the specific architectures and high-level algorithms which allow those services to be provided. For example, one can imagine a group of stationary users time-sharing an infostation in the passenger lounge of an airline terminal, or a single user downloading or uploading messages while passing an infostation. In some instances appearance of a user at an infostation is random and opportunistic, in alternative scenarios, it can also be planned by tracking the user or through knowledge of a route. Each of these assumptions has impact on the strategy for delivering and/or storing the information. The speed of the user places further requirements on this strategy, for if one has only a second to deliver a large file, some form of preparatory communication is probably required. Thus, the infostation might need to exchange low bit-rate information for setup purposes at a somewhat greater range than is provided for the eventual high speed data transmission.

The coordination of the infostations with a backbone network such as cellular gives rise to further issues of architecture and software protocols. The urgency of a delivery will vary, so that optimal strategies for delivery at minimal cost within some time constraint may be complex. How and where to store and replicate information will also be an important issue. The whole "conditional" protocol suite is currently being developed. Conditional e-mail, and ftp (file transfer protocol) wait for suitable communication conditions (such as bit rate, lower error rate, shorter distance to minimize transmission energy) to occur before sending e-mail or transferring a file. This protocol stack supports infostation architecture as a special case.

Finally, the infostation will require a suitable radio. One option would be to use the existing "cellular" radio in a different mode (e.g., using all time slots of a TDMA channel to achieve higher data rates). An alternative is to design a separate radio for this purpose. Our work will focus on a separate but very low cost radio technology together with a specialized layered protocol optimized for this application. Because of its low cost at the user terminal, one can envision a device that most people carry, and which has the potential to support a wide variety of new applications (security badge, credit/ATM/toll payer card, personal data bank). It could serve as a stand alone terminal for short messages, or could plug into a laptop, PDA or some other novel device for storing and displaying more complex data. The coordination of this simple device with a number of different "information appliances" creates an interesting array of human factors issues.

The radio technology we propose is a modernized version of the turn-of-the-century "crystal radio". The antenna, which at high frequencies may be a simple metalized pattern on a suitable substrate, receives an amplitude modulated carrier from a nearby infostation. Digital modulation of this type is referred to as Amplitude Shift Keying (ASK) or On Off keying. If the antenna is small, it will be highly resonant and will provide a degree of frequency band selectivity with no additional filtering[21]

This technology is also capable of providing a very low cost uplink, albeit at a lower data rate than the downlink. The uplink is achieved through "modulated backscatter", in which an unmodulated carrier from the infostation is modulated by the mobile and reflected back to the infostation. Most objects will reflect, or backscatter, an RF signal (as evidenced by radar).

We are currently in the process of developing the architecture for the system and protocol and radio design. This is perhaps the most ambitious of all of the projects since not only it involves a complete design of the protocol stack but building radio as well.

2.4 Direct DB - Satellite Information Systems

This project reflects our early interest in new wireless data dissemination methods.

The system architecture of the *DirectDB* project assumes an asymetric, satellite based communication environment, where a GEO satellite with a very broad coverage downloads information to a large number of clients possibly equipped with the GPS. The uplink channel (from the clients to the satellite) is mainly carrying queries and possibly sensory updates. The downlink channel carries a bulk of information such as maps, visual data (aerial photographs for example) etc. The clients are often disconnected and are not assumed to contiguously listen to the downlink channel. We assume that TCP/IP protocol suite is supported in this environment.

By a *satellite information system* we mean the system in which a database server is attached to a broadcast communication channel. Data from the database is offered to clients in two basic forms: spontaneous, possibly periodic, broadcast (such as broadcast TV) and the demand driven form (i.e. per request, such as standard client server). The former mode has been called *publishing mode* [11] as well as "disk on air" [17], since information is periodically broadcasted as if placed on the rotating disk. Earlier work on publishing mode in wired environments has been described in [7] and [8].

In [11], [18], [19], [12], [16] we have investigated issues of data organization, addressing and scheduling in a situation when the server is periodically broadcasting information and clients are listening to the broadcasted downstream, and filtering information which is relevant to their needs. Under this scenario data broadcast appears to clients as "air memory", or "disk on air" - in fact in [12] we have argued that broadcasting can be viewed as another level of memory hierarchy between the local client's memory and the remote server's memory. In the work just referenced we view data broadcasting as a form of caching on air - data which is expected to be most frequently requested, if not broadcasted, is put on the broadcasting channel. Broadcasting reduces the number of uplink requests for information, since data arrives anyway without being explicitly asked for. However, data on the broadcast channel has to be suitably organized and addressed to allow clients efficient filtering of the incoming data stream to avoid "listening all the time". In [16] we have distinguished two basic methods of addressing information: by using timestamps of information arrival and by using multicast addressing. Additionally, the information which is more important and which is expected to be of interest more often is broadcasted more frequently. Finally, in [18] we have provided a family of methods for including information about changes and updates in an efficient way so even clients who were disconnected temporarily will be able to synchronize their caches.

In the *DirectDB* architecture, the server has two channels: narrow band uplink channel and wide band downlink channel. The downlink channel is further subdivided into logical subchannels, each corresponding to a multicast address.

These logical subchannels are used as follows:

- Possibly periodic, spontaneous (without explicit requests) data dissemination.

  A data item is assigned a multicast address. The client who is interested in downloading that data item has to join multicast group corresponding to that data item.

- Dissemination of data items which were explicitly requested.

  Clients explicitly request data items on the uplink channel. The server does not reply immediately but rather waits to "batch" requests for the same data item together. Then the data item is multicasted on the downlink channel on the predefined multicast address. Thus, multicast addresses are used in the same way as in publishing mode - but this time in reaction to explicit requests.

- Query subchannels

  A query can be mapped to a multicast address, so each record in the answer to this query is multicasted on this address. Thus, here not a single record but a *set* of records is mapped to the same multicast address. A query answer may be broadcasted once, several times or periodically.

- Metadata

  The metadata channels provide the directory information (mapping between the data items and multicast addresses) as well as cache invalidation information for data which changes. Metadata channels help the client decide whether to explicitly request a given data item or to wait for the spontaneous broadcast. We explain the cache metadata in more detail further in the proposal.

The basic problem which we propose to solve is how to distribute the fixed total resources (the downlink channel capacity expressed as bit rate) between the logical channels. Notice, that the number of logical subchannels is essentially unbounded. For example, if 10 Mbit postscript file has to be broadcasted every 10 seconds on a given multicast address m - we need to allocate 1Mb/s to the m logical subchannel. Thus, if the total downlink channel is 24Mb/s, we can afford only 24 logical subchannels meeting the requested service guarantees (in this case the guarantee that the client would not have to wait longer than 5 seconds on average for the broadcasted postscript file). However, for smaller files and less frequent retransmissions we can assign more logical subchannels.

Scheduling can be based on the anticipated rate of request for a given data item. The more frequently requested data items will be broadcasted more frequently [16], [17]. Additionally, as [16] explains scheduling will be based on the priority levels as well as on the quality of service guarantees such as access time. In general, a *scheduling specification* will be expressed as a set of rules describing priorities and desired frequencies of broadcast. This specification will then be translated into a specific policy of data dissemination.

Clients will follow the protocol which we call "Listen First" - by listening first to the metadata channel and determining which information is broadcasted and which is provided on demand.

The decision whether to wait for the broadcast (in case the requested data item is broadcasted) or submit an uplink request depends on the delay which the client is able to tolerate and is a parameter which can be decided by the client. The server maintains queues for each logical channel. Each logical channel, in turn, is mapped to time slots on the downlink channel, depending on the allocated bit rate.

2.4.1 Addressing and Filtering

Solution which we propose are based on low level addressing (network layer). This is beneficial as compared to the application layer addressing which keeps CPU busy examining unwanted packets.

We distinguish two levels of addressing: *multicast addressing* and *temporal addressing*. The mulitcast addresses correspond to the logical channels. The client, upon listening to the metadata, can determine which information is published and which is provided on demand. If the information is published he will joining a particular multicast group which corresponds to the given data item or a query. If it is provided on demand, the proper uplink request will be submitted. In temporal addressing, the timestamp of the next broadcast of the given data item is provided and the client has to wait till that time to download.

Addressing Based on the Client's Location

Location is an important special case of data attribute which is a likely candidate to be used in data filtering.

In [10] we have proposed a way of delivering IP datagrams to the specific geographic locations defined as polygons specified as sequences of GPS coordinates. A similar, though much simpler, scheme may be used to deliver information which is only of interest to users who are located in a particular area. By a *GPS square* we mean a pair of GPS coordinates. Such a pair, using commercially available GPS system defines square of approximately 0.1 mile x 0.1 mile but using a military versions could be much more precise perhaps to the level of a few square meters. Our solutions are based on mapping the geographic areas into multicast addresses. A simple idea is to map individual GPS squares into multicast addresses and have clients, on the basis of their GPS readings join the multicast group corresponding to the GPS square in which the client currently resides in. The server can now use the same multicast address to disseminate information which is specific to the particular GPS square (such a local map). This addressing scheme, however, does not scale well is the information has higher location granularity. Indeed, in case the multicasted information is specific to a wider area than just a single GPS square it would have to be multicasted on a multiple mcast addresses for all single GPS squares contained in the target area. An alternative scheme uses dynamic mapping between the target areas and the multicast addresses. Assume that the server wants to target an area defined by some complex polygon of GPS coordinates. It first multicasts on some well known mulitcast address the "directive" that all clients which reside within the target polygon have to join a specific mulitcast group. Then it mulitcasts the given information on that mulitcast address.

Our ideas of delta publishing can be used to monitor changes in information targeted to geographic areas to avoid rebroadcasting the same large piece of data several times.

For example assume that large aerial map together with enemy positions have been multicasted at 8:20 PM. A given client has downloaded this map but has been disconnected for 45 minutes from 8:45 PM till 9:30 PM. Upon reconnection, it is critical for such a client to determine if no information was lost. For example, it could have happened that a new map with new enemy positions have been multicasted while the client was disconnected. Delta publishing can be used to help the client to quickly synchronize its cache. For example, the timestamp of the last change of the map can be mulitcasted on the very frequently on some special "delta publishing" channel. In this way the client, after reconnection can listen to the delta publishing channel and determine if it needs to request the new map from the server through the explicit uplink request, or if the cached map is still valid.

In general, *disconnection* will be a feature not a bug in the wireless environment and it will be a major factor that will affect caching solutions [18]. In [18] we argue that solutions which require the server to maintain the precise state of the clients, including information who is disconnected, are impractical due to mobility of the clients. Thus, we have advocated a *stateless* solution in which the server does not maintain any information about the state of the clients but rather periodically multicasts an *invalidation report* which, for each data item provides the last timestamp when this data item has changed. The granularity of invalidation report could be much higher than individual cached data items. For example, one entry in the invalidation report can correspond to all NorthEast bound flights of a particular airline. Invalidation report entry (NE, 11:05AM) will mean that the last change to the North East schedules of that airline occurred at 11:05AM. Thus, the clients who were disconnected, say, at 11:30AM and had a copy of the cache with timestamp of 11:15 AM may avoid energy consuming uplink request.

A new set of issues come up when *location dependent data*[3]is cached. Here, cache invalidation may be caused by client's *movement*. Assuming that the client caches several location dependent data items, we would like to avoid a situation where each move by the client would cause a "cache refresh" request to be sent uplink by the client. Different data items may be more or less sensitive to the location of the user. We propose to attach the notion of *scope* to each location dependent data item. This scope is downloaded by clients together with the data items and tells the client in which area the given data item is still valid and when it needs to be refreshed.

2.5 Evaluation Approach

The proposed system has a variable and unbounded number of logical subchannels. The role of the *scheduler* is to allocate resources to these channels (bandwidth) which is equivalent to the decision how often a given data item is going to be broadcasted by the server. Additionally, the server has to make a decision which data items will be broadcasted and which will be provided on demand. Finally, it also has to choose a method for cache invalidation. All these decisions influence the overall performance of the system which is measured though the following set of parameters:

- The average access time

  The average time it takes a client to access a given data item: in case the data item is published it is the time the client has to wait until the given data item is multicasted by the server.

- The average tuning time

[3]Such as a local map

The average time the client's CPU is busy examining packets. This should be measured in terms of "unwanted" packets. The less unwanted packets, which the client has to look at the better "tuned" the system is. This is dependent on the granularity of multicast addressing, specifically how much of the data filtering will have to be performed on the application layer.

3 Conclusions

We have presented here an overview of the projects at the Mobile Computing Laboratory at Rutgers University. This choice represents our point of view on the challenging issues in mobile computing. The wireless web projects deal with improving performance of the most popular, standard, Internet services over the wireless medium. Both transport and application layer protocols will have to address the issues of high error rate, possibly frequent disconnection, energy efficiency, mobility and handoffs as well as highly variable communication conditions. The geographic messaging project introduces location as a first class citizen so messages can be routed to the geographic areas and services with limited geographic scopes can be offered. The infostation project offers an architecture in which the tolerable service delay can be traded for lower cost or for higher bit rate. Finally, the DirectDB project deals with data dissemination from a broadcasting satelite. In all these projects mobility, location, wireless medium and energy limitations play a key role.

References

[1] Tomasz Imielinski and B. R. Badrinath "Wireless Mobile Computing : Challenges in Data Management"
In Communications of the ACM, October 1994.

[2] B. R. Badrinath and Pradeep Sudame " To send or not to send: Implementing deferred transmission in mobile hosts" To appear in ICDCS, Hong Kong, May 1996.

[3] Ajay Bakre and B. R. Badrinath "M-RPC: A remote procedure call service for mobile clients" In Mobicom 95, Berkeley, November 1995.

[4] "Mobjects: Programming Support for Environment Directed Application Policies in Mobile Computing"
B. R. Badrinath and Girish Welling ECOOP'95 Workshop on Mobility and Replication, August 1995

[5] Ajay Bakre and B. R. Badrinath "Handoff and system support for indirect TCP/IP"
In the Second Usenix conference on Mobile & Location independent computing, Ann Arbor Michigan, April 1995

[6] Samuel Sheng, A.Chandrasekaran and R.Broderson, "A Portable Multimedia Terminal for Personal Communications", IEEE Communications Magazine, pp 64–75, December, 1992

[7] T. F. Bowen et.al., “The DATACYCLE Architecture,’ ’ Comm. of the ACM, Vol 35, No. 12, December 1992, pp. 71 – 81.

[8] G. Herman et al., “The datacycle architecture for very large high throughput database systems,” in Proc ACM SIGMOD Conf., 1987, pp 97-103.

[9] T.Imielinski and H.Korth, editors, “Mobile Computing”, Kluwer Publishers, 1996

[10] T.Imielinski, J.Navas, “Geographic Messaging” Request For Comments, Draft, March 1996

[11] T.Imielinski, S.Viswanathan and B.Badrinath, “Power Efficient Filtering of Data on Air”, Fourth Int Conf on Extending Database Technology - March 1994.

[12] T.Imielinski, S.Viswanathan and B.Badrinath, “Data On Air - Organization and Access”, IEEE Transactions of Knowledge nd Data Engineering, to appear in August, 1996

[13] T.Imielinski, H.Korth, “Introduction to Mobile Computing” in “Mobile Computing” edited by T.Imielinski and H.Korth, Kluwer Publishers, 1996

[14] T.Imielinski, B.R.Badrinath, “Data Management Issues in Mobile Computing” in Communications of ACM, October 1994

[15] R.Katz, “Wireless Adaptive Information Systems” in IEEE Personal Communications Magazine, 1995

[16] T. Imielinski and S. Viswanathan, “Wireless Publishing - Issues and Solutions” in “Mobile Computing” edited by T.Imielinski and H.Korth, Kluwer Publishers, 1996

[17] S.Acharya, M.Franklin, S. Zdonik and R.Alonso, “Disks on Air” in ACM SIGMOD 1995

[18] D.Barbara, T.Imielinski, “Sleepers and Workoholics - Cache Invalidation Strategies for Mobile and Wireless Environments” in ACM SIGMOD 1994

[19] T.Imielinski, S.Visnathawan, B.R.Badrinath, “Indexing on Air” - ACM SIGMOD 1994

[20] R.Frenkiel, T.Imielinski, “Infostations - islands of high bit rate”, draft, Rutgers University, March 1996

[21] J.Evans, “Backscatter Radio Technology and Infostations”, draft, Rutgers University, March 1996

[22] A.Bakre, “Indirect Protocols”, PhD thesis, Rutgers University, March 1996

[23] A.Bakre and B.R.Badrinath, “Indirect Transport Layer Protocols for Mobile Wireless Environment”, in “Mobile Computing”, edited by T.Imielinski and H. Korth

[24] A.Bakre and B.R.Badrinath, “ A remote procedure call service for mobile clients” In Proceedings of the 1st ACM Conference on Mobile Computing and Networking, Nov, 1995

[25] R.Katz and E.Brewer "The Case for Wireless Overlay Networks", In "Mobile Computing" edited by T.Imielinski and H.Korth, Kluwer Publishers, 1996

[26] C.Perkins "Mobile IP - request for comments, IETF 1995"

[27] S. Visnathawan, "Publishing in Wireless and Wired Environments", PhD thesis, Rutgers University, March 1995