データベースシステム 100－3
（1994．10．13）

ADAPTIVE WIRELESS INFORMATION SYSTEMS

T. Imielinski S. Viswanathan
Dept of Computer Science
Rutgers University
New Brunswick, NJ 08903

Abstract

We provide an architecture for future wireless information services as a collection of autonomous cells which offer information services of widely varying geographic scope. Each MSS (Mobile Support Station) has the freedom to decide how to allocate different information services to its bandwidth and in which form to offer them. Two basic forms of delivery of information service are identified : periodic publishing mode on a publication subchannel and on-demand mode on the on-demand subchannel. We describe, how each MSS makes the decision about which form of service delivery to use and how to structure its publication subchannel by varying the frequency of publication of individual data items depending on the frequency of requests. In this way we improve *packet efficiency* and *power efficiency* of the client-server protocols by maximizing the number of *queries per Hz* and *queries per Watt*. Each mobile client has to be able to interoperate in this environment. This is assured by making cells *selfexplantory* about their service allocation.

We describe how the presented concepts can be implemented by proposing an extension of WWW protocol, called *adaptive* WWW. We also show how to implement the proposed protocol along the lines of CDPD.

1 Introduction

Wireless technology is gaining popularity very rapidly. Many predict a new emerging, gigantic market with millions of mobile users carrying small, battery powered terminals equipped with wireless connection. Mobile users will be in constant need of information such as traffic information, local directory, weather information, local sales, etc. In this paper, we discuss the general issues and provide some specific algorithms to design future wireless information services.

Wireless Information services will be characterized by their geographic *scope*. *Wide area* services such as Stock Market Information will be offered on a national scale. *Macroservices* such as weather would be provided on a regional basis: regions extending to tens, if not hundreds, of miles. The geographic scope of *Microservices* such as traffic conditions will range through the areas comparable in size to the current cells: a few miles in diameter. Finally, one could imagine *picoservices* in an area corresponding to future *picocells*: a few hundred meters in diameter. For example, parking lot availability is the information that can be provided within this scope. We assume the hierarchical cellular organization with macro, micro and possibly picocells. The granularity of services may not directly correspond to the cell granularity (macrocells, microcells) and should not be confused with it.

Wireless Bandwidth and *Battery Power* on the client side will be two most scarce resources

which will have to be efficiently handled by the proposed solutions. This translates to *packet efficient* and *power efficient* solutions, as described later. *Mobility of the clients* and their interoperability with different cells and different air interfaces will have to be actively supported by designing suitable *handoff* protocols. Finally, new families of services such as active services (like triggers) will have to be supported. In general, our goal is to be able to adequately support *existing Internet services* as well as provide ways to support new services. With new significant resource restrictions this is a challenging task.

In the next section, we describe the physical system environment, define an abstract concept of an information service and sketch the proposed solutions. In section 3, we describe the publishing mode and show one example of organization of the publication channel and demonstrate how in that particular case the decision is made about "publishing" a data item and the publication frequency. In section 5, we describe the adaptive algorithm for providing the information services. Later, we follow it up with the general model of a server and describe a possible wide spectrum of solutions in which the server (MSS) uses different scheduling algorithms to accommodate possibly widely varying performance criteria. Section 6 provides a direction for the implementation of the proposed concepts and finally section 7 concludes the paper.

2 Basic Notions

We begin with the sketch of the underlying system architecture.

2.1 Systems Architecture

The mobile wireless environment consists of the set of *wireless cells* possibly of varying sizes. Current cell sizes are few miles in diameter but, in the future, cells may have much smaller size (so called *picocells*) due to the better frequency reuse. Each cell is managed by a base station called *Mobile Support Station* (MSS). There are outdoor and indoor cells. Cells may vary widely by the type of the wireless links which they use. The indoor cells may use higher bandwidth wireless LAN such as Wavelan or range LAN as well as infrared based LANs. The typical bandwidth on wireless LANs ranges from 1 Mb/s to possibly a few Mb/s for the future products. The outdoor cells typically have and will continue to have far more restricted bandwidth. Current solutions based on CDPD provide 19.2 Kb/s. With the recent allocation of new bandwidth for PCS (Personal Communication Services) by FCC, the situation may improve, but it is reasonable to assume that the outdoor bandwidth will be more scarce than the indoor one. Therefore, cells will vary both in terms of size as well as in terms of wireless links. Tariffs will also vary, ranging from the fees based on connection time to the fees which are proportional to the number of packets sent/received.

Figure 1 shows the structure of a hypothetical system which provides mobile users with information services. The wireless information servers (denoted by tall antennas) will be equipped with wireless transmitters capable of reaching thousands of clients (denoted by small dark rectangles) residing in macrocells[1]. Clients equipped with small palmtops with wireless connections, will move freely between cells and query information provided by the wireless information servers [12].

Wireless Information Services

Most of the current wireless information services are "receive only." Only recently, new services are emerging which in fact offer two way wireless communication. Motorola's Embarc is one of the first such information services provided on the market. The user equipped with the HP95 LX palmtop and Embarc's pager can receive and buffer news services including newsbites, financial services, etc, by satellite

[1] as opposed to the microcells which are expected to facilitate applications such as e-mail, macrocells of a bigger size will be used for wider area information services

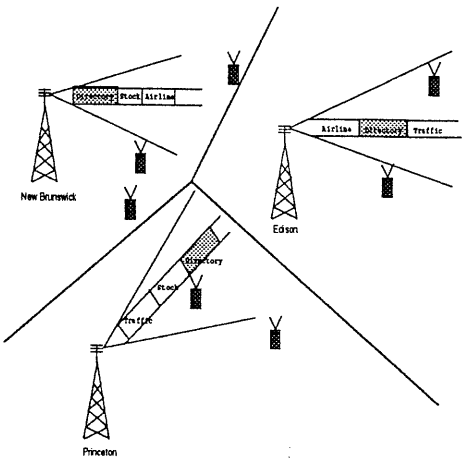

Figure 1: Wireless Information Servers

using one of Motorola's paging channels. Information is multicasted on a specific address which is matched by the client's terminal.

Radiomail provides wireless e-mail capabilities in the areas covered by RAM Mobile network. ARDIS, a joint venture of IBM and Sears offers numerous wireless services in a number of metropolitan areas. There is a number of applications dealing with a particular niche market such as navigation and information services for a truck fleet (such as Quolcomm's system). Gifford in [6] describes a community information system which uses FM band to broadcast wireless information to PCs equipped with wireless modems.

In general, however, these are isolated and application specific information services. There are still far away from the vision of using existing Internet services and defining new services for wireless and mobile environments.

Future Wireless Information Services :
Below, we list a few examples of possible future wireless information services:

- **Picoservices**
  - Local Parking Lot availability
  - Layout of a Building/factory
  - Empty seats in a stadium/ restaurant/ concert hall
  - Items in a store
  - Shopping mall advertising[2]
- **Microservices**
  - Airport Information
  - Train and Bus Station Information
  - Local Traffic Information
  - Local Directions
- **Macroservices**
  - Weather Information
  - Regional Maps
  - Local News Services
  - Local Telephone Directory Services
- Wide Area Services
  - Stock Market Information
  - General News (financial, sports, commodity prices etc)

These services may be offered in many different forms. For example, active services may enable the user to be alerted when some particular conditions become true. Stock reaching a particular level, traffic congestion on a particular street, a plane departure delayed can all be examples of events which a mobile user may be alerted about.

Motivation for Energy savings :
The lifetime of a battery is expected to increase only 20% over the next 10 years [20]. A typical AA cell is rated to give 800 mA-Hr at 1.2 V (0.96 W-Hr). The constraint of limited available energy is expected to drive all solutions to mobile computing on palmtops.

[2] A user who is passing through a shopping area may receive advertising information about local sales and events which are *local* with respect to his/her current location.

Assuming that the power source of the palmtop to be 10 AA cells with a CD-ROM and a display, the constant power dissipation in a CD-ROM (for disk spinning itself) will be about 1.0 W. The power dissipation for display will be around 2.5 W. Thus the assumed power source will last for 2.7 Hrs. Thus to increase the longevity of the batteries, the CD-ROM and the display may have to be powered off most of the time . Apart for CD-ROM and display, the CPU and the memory of the palmtops also consume power.

There is a growing pressure on hardware vendors to come up with the energy efficient processors and memories. The Hobbit chip from AT&T is one such processor which consumes only 250 mW in the full operation mode (*active mode*). The power consumption in *doze mode* is only 50 μW (the ratio of power consumption in active mode to doze mode is 5000). When the palmtop is listening to the channel, CPU must be in active mode for examining data packets (finding, if they match the predefined data). CPU is a much more significant energy consumer than the receiver itself and since it has to be active to examine the incoming packets it may lead to waste of energy, especially if on an average only a very few data packets are of interest to the particular unit.

Therefore, it is definitely beneficial if palmtops can slip into *doze mode* most of the time and come into *active mode* only when the data of interest is expected to arrive. This requires the ability of selective *tuning*, which is discussed in detail in [13] and [14].

The power needed for transmission from the mobile client to the MSS is proportional to the fourth power of the distance between the two. As the distance increases, the power required is very high. Moreover, a number of factors like the terrain, the landscape, the height of the trees, the foliage, the kind of trees, the season, rain, etc, play an important role in the determining the power required by the client in transmitting to the MSS [21]. Power being one of the scarce resources in the mobile client, it would be beneficial if the client can avoid transmission to a lot of extent and still get the required data.

Energy efficient solutions are important due to the following reasons :

- Energy efficient solutions make it possible to use smaller and less powerful batteries to run the same set of applications for the same time. Smaller batteries are important from the portability point of view since, palmtops can be more compact and weigh less.

- With the same batteries, the unit can run for a very long time without the problem of changing the batteries ever so often. This can result in substantial monetary savings and avoids recharging. Recharging can be cumbersome especially if the client is on the move. With energy efficient solutions batteries may have to be recharged only every few days, rather than every few hours.

- For environmental consideration. Every battery that is disposed is an environmental hazard.

In the next subsection, we describe more formally the concept of information server which we are going to use in the rest of the paper.

2.2 Information Servers

An *information server* is a collection of pages which are connected by *links* as in World Wide Web (WWW) protocol. In WWW protocol each page has its address which is a pair consisting of a port number and an IP address of the server offering this page. *Mosaic* and *Gopher* are both graphical interfaces to WWW. In Mosaic, links are associated with individual words of the page. URL (Universal Resource Locator) provides a mapping between pages and their addresses. Below, we show the two

top pages of the Rutgers University's INFO system implemented using gopher.

INFO: Rutgers University Pilot Campus-Wide Information System

Each menu entry is an IP address of a server which may, in turn provides a tree of services structured as a sub-menu. The leaves of the menu tree provide specific information pages such as documents, telephone numbers etc. For example, the *About_Rutgers* entry in the top level menu will lead to another menu asking the user to select a sub-menu corresponding to information about the university, Calendar of events, the directions to the university etc.

Command	Purpose
About_Rutgers	General Information about the University
Academics	Courses, Schedules, Registration, Special programs
Camden	Activities and events on the Camden campus
Computing	Computing facilities, services and network information
Directories	Phone directories, Faculty research info., Univ. Forms
Using_INFO	<<What is INFO, how to use it, how to send suggestions>>
Library	Libraries, information resources, reference material
Newark	Courses and events in Newark Campus
News_n_Events	News, Weather
Services	Students,Faculty/Staff services, police info
University	University , Community

General Information about the University Menu Commands...

Command	Purpose
Rutgers	About the University
Calendar	Academic calendar and staff holidays
Directions	Directions to various campuses
View_Maps	Maps for campuses - view on X workstation
FTP_Maps	Maps for campuses - where to get online
Recruitment	Rutgers presence at Recruitment Events

WWW and its graphical interfaces are extremely popular on the fixed network. In this paper, we would like to investigate the possibility of extending the use of WWW like services on wireless networks for mobile users carrying small battery powered terminals.

Let us make a distinction between two different types of activities of a user of an information service.

- Browsing

  In browsing activity the user is traversing a hypelink style structure from page to page. The directory of information is fully integrated into the data representation - individual links between pages are associated with addresses.

- Ad Hoc, Free Querying

  In ad hoc querying, the user is interested in specifying a keyword or a value without browsing through the complex hyperlink structure. Here, the explicit form of directory is necessary to provide the user with the proper address of the requested data item.

We argue that WWW protocols in its current form are not appropriate for the mobile and wireless environments since they are neither *packet* nor *power* efficient. By *packet efficiency*, we will mean the ability of the protocol to

minimize the total number of packets sent on wireless link. By *power efficiency*, we mean the ability to minimize the energy spent by the battery powered client using this protocol.

The WWW protocol is not packet efficient since, regardless of the number of requests for a particular page, the server will download the page in response to each individual request. A better solution would be to periodically multicast such a page to avoid a large number of uplink messages as well as a large number of downloads of the same page. Additionally, each page requires the wireless client to make a separate TCP/IP connection with the server using a 3-way handshake protocol. In this way many extra packets are sent on the narrow wireless link.

The WWW protocol is not power efficient for a very similar reasons that it is not packet efficient. Uplink requests require energy consuming transmissions which are particularly hard on batteries when the distance between the mobile station and the MSS grows. Periodic multicasting seems to be a good solution in both of the above cases, since the client is only required to listen to the channel without expensive uplink transmission and one packet can satisfy a large number of requests.

Queries per Watt?, Queries per Hz?

In cellular telephony one of the most important performance criteria is the cell capacity: the number of telephone calls which may be handled by the cell per unit of time. In the future - a *query* - a single request for a page will be an analog of a telephone call. Consequently, we believe that future cell capacity will be measured as a number of *queries* which the local MSS can handle per unit of time as well as a number of queries which a mobile client can issue before it runs out of batteries. This leads us to two major performance indexes: *queries per Hz*[3] and *queries per Watt* which measures a number of queries which a given client can process from a 1 Watt battery. We propose an algorithm in which periodic multicast of a subset of pages (data items) will help us in improving both performance criteria : *queries per Hz* and *queries per Watt.*

Two modes of delivery - instead of one

The communication channel consists of a downlink channel and an uplink channel. Each wireless cell will have a choice of two basic forms of information dissemination and clients access information using the communication channel in the following two modes:

- *Publishing Mode*: Data is multicasted periodically, on the downlink channel. Accessing multicasted data does not require uplink transmission and is "listen only." Querying involves simple *filtering* of the incoming data *stream* according to a user specified "filter" [4].
- *On-Demand Mode*: The client requests a piece of data on the uplink channel and the server responds by sending this data to the client on the downlink channel.

While the second mode of interaction is the traditional client-server approach, the first mode is different [5] - here the server either periodically broadcasts information or just sends it on a specific channel (say a multicast address). This is analogous to the way information is disseminated via newspapers and regular TV broadcasting. Such published data will eventually be filtered by the client using a directory which is provided along with the data, by the server. There are many advantages of the publishing mode: broadcasting the most frequently accessed data items (hot spot) saves bandwidth since it cuts down the number of separate but identical requests, therefore it is *packet efficient.* It also saves client's energy by avoiding energy consuming uplink transmissions and thus is *power efficient.* Providing the directory helps the client to selectively tune only to the relevant information and remain in the *doze mode* for the rest of the time. This

[3]Maximizing the number of queries which the wireless link with a given bandwidth can handle in a unit of time

strategy saves considerable amount of energy as demonstrated in [13] and [14].

In terms of bandwidth allocation, we may view each cell's bandwidth in terms of three channels: the *uplink* channel, the *on-demand downlink* channel and the *publishing downlink* channel. How much bandwidth is allocated to each channel is cell dependent and is a function of the cell population and the pattern of requests in each cell. One cell may carry or "syndicate" stock market information on its broadcasting channel, another cell may decide, due to the limited interest, to provide this information only on-demand. The allocation of information services to the physical bandwidth will vary from cell to cell pretty much in the same way as radio and TV programs are mapped to different frequencies in different geographic areas. The analogy with TV stations can be carried even further by treating the publishing (broadcasting) channel of the MSS as its *prime time* information offering. This decision will be based on the frequency of requests which is analogous to the concept of *TV rating*. However, by contrast with local TV stations, which may not "carry" certain TV productions, MSS will usually provide (on-demand) access to all public domain information services on Internet (assuming it provides Internet access). Services which require high bandwidth such as some of the Mosaic based multimedia services may have to be presented differently on narrow bandwidth cells. Moreover, the MSS may decide not to carry some of the commercial services or may only carry them in a very limited fashion.

Cell Autonomy

In general, cells will differ in the type of radio infrastructure and consequently by the amount of available bandwidth. Consequently, the form of presentation and access to a particular service will be *location dependent*. There is an obvious need for a protocol suite which would make this dependency hidden from the user who relocates between such cells. There is a need for *service handoff* in a way similar to, how cellular handoff is handled now in order to assure continuity of voice conversations. Some of the services will have to be continued upon crossing cell boundaries with minimal service disruption. Since mobile clients will cross cells, interoperability with services offered in different cells is of utmost importance. Therefore, each cell should be *selfexplanatory* providing the description of the "directory of the services", their mapping to the physical bandwidth and the form of their delivery: broadcast, multicast, or on-demand. The exact map between services and their addresses will not be visible to the user but simply allow the system to "rebind" the services according to the allocation used in the new cell.

Summarizing, the mobile user will see the same interface regardless of his/her location. Some of the services, though (such as location dependent, pico or microservices), will change along with the location of the user. The "binding" of the services will vary from cell to cell but will be transparent to the user. Client's *interoperability* with individual cells will be achieved by assuring this transparency.

The above described infrastructure of wireless information services poses many important questions: assuming such widely varying environment, how do we structure the information servers to offer a wide area service on several cells? How do we assure transparent access to information resources for the mobile user, i.e., how do we provide the service handoff? How do we provide the data items, so that the clients can access it with minimal energy consumption? How is the bandwidth allocated between the uplink channel broadcasting channel and on-demand channel in a way that *adaptively* reflects the pattern of usage in each cell? We concentrate on providing solutions for the last two questions and give directions in which to proceed for solutions for the other two questions.

We want to avoid changing the software at the server in order to accommodate mobile and

wireless clients in different cells. Such clients may be using different wireless environments by residing in cells with varying air interface. The local MSS should have autonomy to decide how a particular remote service is going to be presented to users registered in its cell. Consider the example of a stock information service: the MSS may decide to publish (broadcast) stock quotes for the S&P100 stocks, every minute, broadcast stock quotes for the S&P500 stocks, every 5 minutes and provide other quotes on-demand with a cache invalidation service [2] which periodically broadcasts reports only about significant (say more than 1 point) changes of the stock values. MSS may also decide that special events such as a stock reaching a new high requires a special broadcast message which should be sent to subscribers. The remote stock server will not, and should not, be aware of this autonomous decision of the MSS. In fact, different MSS may decide to "carry" the stock information services differently in order to best utilize its own resources and perhaps also maximize the profit[4]. Each MSS will know best the local characteristics of its local population and consequently will know which modes of service delivery is the best.

Below, we discuss the publishing mode in more details.

3 Publishing Mode - Energy Saving Access

Before demonstrating the energy saving advantages of the publishing mode let us explain the distinction between the active and doze modes.

Active and Doze modes

We assume that the client's CPU has the ability to operate in at least two major modes: the *active mode* and the *doze mode*. The doze mode describes a low power operation mode where CPU performs only a very limited set of operations. In the AT&T's Hobbit chip the doze mode requires 5,000 times less power than the active mode. By a *low power operation* we mean, an operation which can be performed by the client *in the doze mode*. We are interested in maximizing the use of low power operations in client-server interactions. Below, we list some of the operations which can be provided without involving (waking up) the CPU:

- Filtering by matching multicast addresses:

  Assuming that the client is equipped with an Ethernet Card or a pager, such as the one used by Embarc or the new *Inflo* receiver [5] it can match the predefined set of packet addresses without waking up the CPU and utilizing only the low energy consuming circuits of Ethernet card or the pager.

- Filtering by matching a specific timestamp

  Since clock can be operated as a low energy circuit, matching a predefined timestamp can be performed entirely in a doze mode without waking up the CPU. In this way the client can wake up at the specific time to receive relevant information.

- Storing in a buffer

  Receivers used in pagers such as Embarc and the recently announced *Inflo* are capable of storing up to 120 Kbytes of data (Inflo). This buffer is built as a low energy circuit, which does not require CPU to operate in active mode.

We will structure the client server interactions in a way which maximizes the use of low energy operations and minimizes such energy consuming tasks as client's initiated transmissions. This in turn, will lead to a different structuring policy for a server which, at least

[4]No one knows what the tariff structure will be for the future wireless services but one may assume that the MSS will have to pay the remote server for syndicating its service and will generate a profit by reselling this service to its local residents

[5]Recently announced by Motorola and NewsPager Corporation of America

for the most frequently accessed data items will operate in *publishing mode* rather than in *on-demand mode*. In publishing mode, the server periodically publishes information without waiting for the explicit request for it. Publishing mode is currently used by all major media such as TV, radio and newspapers. Publishing mode requires no transmission from the client, but may require continuous tuning for the client which wants to locate a data item it is interested in. Thus, in order to further minimize the energy it is necessary to provide the client with the ability of *low power filtering* operations. This can be accomplished by making sure that the client remains in the doze mode and wakes up only when the "relevant" data is being published.
Summarizing, the low energy information system will be structured using the following principles:

- The server periodically publishes the most frequently accessed data items. The rest of the data items are provided "on demand" in the usual client-server mode of operation This minimizes the number of transmissions from the client, since accessing published data items requires only "listening."
- The client has the ability of selectively listening to the published information by remaining in a doze mode and waking up only when relevant information is published. This avoids excessive energy consumption on the client's side due to continuous tuning to the published information in the active mode.

The published information can be viewed as a *cache* or cache on the air [15]. The client first tries to locate the item it is looking for on the cache and only upon cache miss, does it submits a request to the server. Hence, transmission from the client takes place only upon cache miss. Additionally, the cache itself is *addressable* in order to enable selective listening on the client's side. The addressing of server's published information should be done in such a way that address matching is a low power operation, i.e., address matching is performed in the *doze mode* without engaging the CPU.
This can be implemented using two basic operations which can be performed while in doze mode:
The publishing channel can be viewed as a virtual memory which is addressable in such a way that address matching is a *low power operation.* Technically, the address of a page on the network is an element of the cartesian product:

$$TIME * CHANNEL$$

The TIME domain is the set of timestamps and the CHANNEL domain is the set of multicast addresses[6]. For example, the IBM stockquote may be published at 11:30 AM on the multicast address 255.6.4.2. For selective tuning it is imperative to provide a directory information to the clients. The directory provides the mapping between the pages and their addresses.

In [13], [14] and [15] a number of methods have been proposed to multiplex the directory (index) information together with the data on the same channel, in case the address refers simply to the time of the publication of a data item. One can extend these methods to the situations when the address is a hybrid of temporal and multicast addressing. The mapping between the data items and the addresses can be performed by a hashing function, which when applied to a data item returns the multicast address.

For instance, in a stock market information service, each stock quote can be mapped to a multicast address. There are 28 bits available for *multicast groups*, which provide an address space of 2^{28} addresses. A hashing function which is known to the receiving client is applied to the name (or its three letter abbreviation) of the stock that it is interested in, to obtain the

[6] In IP, this constitutes all 32 bit addresses starting with 1111.

multicast address. This is the finest possible granularity of indexing which leads to non-efficient use of packets since usually a single packet can carry numerous stock quotes, not just one. In case numerous stock quotes are "packed" into one packet only one multicast address is needed for every (such) group of stocks. Hashing into *buckets* rather than single addresses can accomplish this goal.

The client then, joins such a multicast group and is able to filter the relevant packets using Ethernet card only, without waking up the CPU and the whole IP protocol stack. This protocol has actually been implemented and the details are reported in an upcoming technical report. It also describes how more complex events (triggers) such as "New High" or stock going up by more than a predefined threshold can be implemented.

3.1 Publishing Algorithms

Clearly, there are a number of ways of structuring the publication of a set of data items (pages). We will first list a few possible ways and then discuss one of these in detail.

Publishing is simply a periodic downloading of data items by the MSS. As discussed earlier, data items can have addresses which can be either multicast addresses, temporal addresses or combinations of both. Given a list of data items to be published and their addresses the MSS will have to act essentially as a *scheduler* to decide which items to download in the next frame. Temporal addressing introduces additional very specific constraints in which the MSS must simply download a particular page at a specific time (or in case of a more flexible temporal addressing *within* a particular temporal interval). Additionally, the questions of *directory publication* have to be addressed. In the browsing mode, the directory is built in into the hyperlink structure of the server, where the pointers are simply associated with links. To accommodate a free style, ad hoc mode of querying, the directory has to be explicitly published as well. In [13], [14] and [15] we discuss in detail how the index information (directory) can be multiplexed together with the data on the downlink channel in case, temporal addressing is used. For multicast addressing, a hashing function which maps the keys of data items (or pages) to the multicast addresses has to be provided to the clients.

Since the hashing function takes much less space than the full directory it can be provided to the clients on a "one to one" basis upon crossing a cell boundary (as part of the registration).

In this paper, we concentrate on the multicast addressing. The proposed algorithm for *publication scheduling* is one instance from a possibly broad range of protocol. It is motivated by differences in the relative frequency of page access. In the proposed method, the pages are published depending on their demand[7]. The more popular pages are published more frequently than the less popular ones.

Consider k pages $D_1, \ldots, D_k$, all of the same length S which are requested with the frequencies $\lambda_1, \ldots, \lambda_k$. We assume that published pages are assigned to multicast addresses using some predefined hashing function. In the following we concentrate on the decision mechanism for the MSS to output the next page. We describe a randomized algorithm for minimizing the access time[8] for the pages. In the randomized algorithm that we suggest, the page that will be published next will be determined by *a random function.* This *random function* μ is determined apriori, based on the demand for the various pages. Let $\pi(i), i = 1, \ldots, k$, be the probability that page i is demanded in a unit time slot, $\pi(i) = \frac{\lambda_i}{\sum_{j=1}^{k} \lambda_j}$.

Initially, the most frequently page is chosen to be published in the publishing channel.

[7]Section 5.1 describes some of the ways of gathering statistics regarding the demand for various pages.

[8]The *access time* for a data item is defined as the time elapsed from the point the clients wants the data item to the point when it retrieves the same.

After the multicast of a page, the next page to be multicasted is determined by the random function μ. The multicast of each page can be considered as a *slot* in the publishing channel and the page that will be multicasted in that slot will be determined by the function μ.

The access time of a page will be determined by the function μ. A judicious choice has to be made for selecting μ so that the access time is minimum. The probability that a data item D_i is scheduled in the next publication[9] is $\mu(i)$. The probability that data item D_i is not scheduled in the next publication is $(1-\mu(i))$. If D_i is not scheduled to be published in the next slot, the access time for this page is increased by the duration of the publication for the page scheduled in that slot. The duration of the publication of a page of size S, on a publication channel with bandwidth B_b is equal to $\frac{S}{B_b}$ units of time.
If D_i is scheduled in the j th slot from the current slot then the access time for this item is

$$(1-\mu(i))^{j-1} * \mu(i) * j * \frac{S}{B_b}$$

Thus, the expected access time t_i for an item D_i is:

$$t_i = \sum_{j=0}^{\infty}(1-\mu(i))^{j-1} * \mu(i) * j * \frac{S}{B_b}$$

$$t_i = \frac{S}{B_b} * \mu(i) * \sum_{j=0}^{\infty}(1-\mu(i))^{j-1} * j$$

Using elementary calculus, $\sum_{j=0}^{\infty}(1-\mu(i))^{j-1}*j$ can be reduced to $\frac{1}{(\mu(i))^2}$.

$$t_i = \frac{1}{\mu(i)} * \frac{S}{B_b}$$

The overall expected access time for all the k pages is :

$$t_b = \sum_{i=1}^{k}\pi(i)\ (\ \sum_{j=0}^{\infty}(1-\mu(i))^{j-1} * \mu(i) * j * \frac{S}{B_b}\ )$$

$$t_b = \sum_{i=1}^{k}\frac{\pi(i)}{\mu(i)} * \frac{S}{B_b}$$

Using the 1 st order optimality condition in Non-Linear Programming, the minimum overall expected access time is obtained when the random function μ is as follows:

$$\mu(i) = \frac{\sqrt{\pi(i)}}{\sum_{j=1}^{k}\sqrt{\pi(j)}}$$

Hence, the minimum expected access time for the pages using the above random function is:

$$t_b = (\ \sum_{i=1}^{k}\sqrt{\pi(i)}\ )^2 * \frac{S}{B_b} \qquad (I)$$

4 On-demand Access Mode

This is the conventional client-server model where the client makes a request to the server for a particular item and the server sends that item to the client. We consider a $1/M/M$ queuing model. If a bandwidth of B_o is allocated for the on-demand service then the expected access time for an item is:

$$t_o = \frac{1}{\mu - \lambda} \qquad (II)$$

where μ is the service rate of the requests and λ is the arrival rate or the demand for some item in the on-demand group. The service rate $\mu = \frac{B_o}{S+R}$, where S is the size of the page and R is the size of the message for requesting the pages. λ is the sum of the arrival rate for all the data items in the on-demand group.

5 Adaptive Access Protocol

Having introduced the basic two modes of information dissemination - *on-demand* mode

[9]next publication refers to the next page which will be downloaded on the publication channel

and *publishing* mode we will now discuss how to use both modes for efficient information access. How do we decide whether a particular page is to be published or provided on demand? How often should the pages (that have been selected for publishing) should be published? Below, we provide one solution and follow it up with a more general discussion describing the whole spectrum of possibilities.

The MSS's pages are partitioned into two groups namely, the *publication group* and the *on-demand group*. The pages in the publication group are periodically multicasted by the MSS on the publication subchannel. The pages in the on-demand group are provided on the on-demand subchannel.

Since different pages may have different access frequencies it makes sense to publish more frequently accessed pages more often. The proposed solution critically depends on the overall performance criterion. In the following algorithm, we are interested in :

- *Minimizing the number of transmissions per query while making sure that the expected access time does not exceed a certain predefined threshold T.*

Clearly, without setting up any threshold on the expected access time the server could simply publish all pages, periodically. No transmission (uplink request) would be necessary from the client and thus with an adequate directory, the overall energy per query would be minimal. The access time, however, could be unacceptably large. Thus, there is a need for a tradeoff - only a subset of pages can be published - the k most frequently accessed ones. The cutoff point k, is chosen as the largest integer such that the overall expected access time for any page does not exceed T.

Scheduling Algorithm for the MSS :
Consider the information server with n pages $(D_1, \ldots, D_n)$ all of the same size S. Let λ_i denote the number of requests for the page D_i per unit time. Arrange the pages in the decreasing order of the demand. Let the resulting sequence be $D_1, \ldots, D_n$.

The idea is to publish pages that are most popular and provide the rest of the pages on demand. This requires the partitioning of all the pages into two groups. One consisting of the pages that are published according to the *Publication Scheduling* and the other group consisting of the pages that are provided on demand. Let an integer k ($1 \leq k \leq n$), be chosen and all the pages: $D_1, \ldots, D_k$ constitute the *publication group* and $D_{k+1}, \ldots, D_n$ constitute the *on-demand group*
The expected access time for any page is:

$$t = \sum_{i=1}^{k} \lambda_i * t_b + \sum_{i=k+1}^{n} \lambda_i * t_o$$

where t_b denotes the access time of an page in the *publication group* and t_o denotes the access time of an page in the *on-demand group*. Substituting the values of t_b (from equation (I)) and t_o (from equation (II)) we get :

$$t = \sum_{i=1}^{k} \lambda_i * \left(\sum_{i=1}^{k} \sqrt{\pi(i)} \right)^2 * \frac{S}{B_b} + \sum_{i=k+1}^{n} \lambda_i * \frac{S+R}{B_o - \sum_{j=k+1}^{n} \lambda_j * (S+R)} \qquad (III)$$

If the overall bandwidth of the communication channel is B. Then, this bandwidth can be divided into two parts: B_b (the bandwidth allocated for publishing channel) and B_o (the bandwidth allocated for on-demand service). Thus, the communication channel is divided into two logical channels: the publishing channel and the on-demand channel.

$$B = B_b + B_o \qquad (IV)$$

Substituting equation (IV) in equation (III) we get :

$$t = \sum_{i=1}^{k} \lambda_i * \left(\sum_{i=1}^{k} \sqrt{\pi(i)} \right)^2 * \frac{S}{B_b} + \sum_{i=k+1}^{n} \lambda_i * \frac{S+R}{B - B_b - \sum_{j=k+1}^{n} \lambda_j * (S+R)}$$

We differentiate the above equation, equate it to zero and evaluate the value of B_b. This gives the optimum way in which the

communication bandwidth has to be split into the publication bandwidth and the on-demand bandwidth in such a way that the access time is the minimum.

$$B_b = \frac{ca + \sqrt{a^2c^2 - c^2a(a-b)}}{(a-b)} \qquad (V)$$

where $a = \sum_{i=1}^{k} \lambda_i * (\sum_{i=1}^{k} * \sqrt{\pi_i})^2 * S$
$b = \sum_{i=k+1}^{n} \lambda_i * (S + R)$
$c = B - \sum_{i=k+1}^{n} \lambda_i * (S + R)$

To minimize energy consumption per query, we have to publish as many pages as possible. It may not be possible to provide the required access time if all the pages are published, hence the most popular pages will be published and the rest provided on demand. Now, we give an algorithm for getting the minimum power consumption per query, given a communication channel of bandwidth of B and an expected access time requirement of T.

ALGORITHM : Consider the sequence $D_1, \ldots, D_n$.

- For k = n downto 1 do:
  begin
  - Classify $D_1, \ldots, D_k$ as the *publication group*
  - Classify $D_{k+1}, \ldots, D_n$ as the *on-demand group*
  - Compute B_b using Formula (V)
  - Compute B_o using Formula (IV)
  - Compute t using Formula (III)
  - If ($t < T$) Then break

  end
- All the items in the *publication group* are published on the channel according to the *Publication Scheduling*.
- All the items in the *on-demand group* are provided in the on-demand mode.

Access Protocol for the Client :
The client upon entering a cell sends a greeting packet to the MSS and in response, receives the address of the root page. The address of the root page could either be an IP address which the client has to connect to in order to receive the root page or in case the root page is published, the publication address for this page. The publication address is a multicast or temporal address or, possibly a combination of both. We assume that each page contains information about the status of its children - whether they are provided on demand (in which case the IP address is specified) or, if they are published (in which case the temporal address is specified). Notice, that once a page is provided on demand, all its children are provided on demand as well. Hence, the client upon first tuning to the publishing channel simply "follows the pointers" in the successively accessed pages. By following the pointer, we mean that if the address of the page is an IP address, then the client will request the MSS for the page else, (the address is a multicast address) the client will download the page from the publication subchannel as follows: The client, first *listens* to the channel, using the multicast address of a requested page. If that page is not provided within a specific period L then the client submits an uplink request.

5.1 Gathering Statistics for Dynamic Scheduling

The adaptive scheduling algorithm described for the MSS is a dynamic scheduling algorithm. Given the frequencies of access for the various pages we can use *publication scheduling* for the publishing group. The key difficulty is how to obtain the frequency of access numbers for the pages which are published rather than provided on demand? Obtaining such *feedback* is contradictory to the nature and

motivation of publishing which assumes no uplink activity from the clients regarding the published material. Gathering statistics for the pages which are obtained on demand is straightforward. Hence, in the following we concentrate solely on statistics gathering for publishing mode.

Different methods depend on how much "up to date" the MSS wants to be regarding the distribution of requests. Here are a number of possible solutions:

- When the client sends a request to the server for accessing a page provided by the on-demand model, it piggybacks an additional message. The piggybacked message will include the number of times each page belonging to the publication group was downloaded between the last access of a page belonging to the on-demand group and the current request.

- If the information service is organized as a tree (WWW) then, it is usually the case that the top few levels belong to the publication group and the rest especially the leaves belong to the on-demand group. Hence, the request for each leaf goes to the MSS. The frequency of access for each of the other pages in the interior of the tree can be extrapolated. The frequency of access of a page (in the interior of the tree) is equal to the sum of frequency of access of its children.

- In case the client informs the MSS just before disconnection (of its impending disconnection), then it can piggyback a message indicating the number of pages of the publication group that it accessed in the duration of its registration under that MSS.

- The demand for published pages can be measured periodically by dropping a page from the publication group and providing it on-demand for a while, thereby estimating as to how frequently that page is requested. This frequency for a page is maintained till the next time that page is dropped again from the publication group. Depending on how accurately the MSS has to be informed of the frequency of access for various pages, the pages can be dropped often or just once in a while.

It has to be noted that while gathering the statistics regarding the frequency of access for a particular page, frequency of access has to be measured as the average of the number of requests for that page from the point it was sent last (either by publishing or on-demand mode) to the current moment. If the frequency of access is measured only locally then, the correct statistics is not obtained. All of the above solutions also depend on the ability of clients to send uplink information (which may not be the case for one way pagers).

Publishing Events

Publishing mode can also be used for publishing *events* which inform the clients about data *changes*. This is very useful in implementation of active services as well as in handling cache invalidations in the presence of frequent *disconnections*. For example, in [2] we describe a protocol in which the MSS periodically publishes an *invalidation report* which for any data item contains the last timestamp when that item changed[10]. The client, determines if its cache is valid by listening first to the published report and going uplink only if the cache validity is no longer guaranteed. The number of uplink requests is reduced and the MSS does not need to maintain any state information about the clients residing in its cell. This method allows for the clients to disconnect and still be able to use their caches provided that the item has not changed and that disconnection was not too long (less than a certain predefined parameter).

The most frequently requested triggers can

[10]Only the most recent changes are included, so if the item did not change for a long time it is not included in the report

be published as well. For example, the delay of a large plane can be published periodically instead of sending separate unicast messages over narrow wireless band.

5.2 General Model of a Server

We can generalize the above discussed concepts of the information server as follows:
There is one downlink channel which consists of *frames*. The MSS at any point of time has to *schedule* the next frame to be downloaded on that channel. For that, the MSS can use a *scheduling algorithm* from a wide range of algorithms developed in the literature [3]. In general, there are two basic queues: the *on demand* queue and the *publication* queue. The MSS's scheduler at any point of time t selects one page to be downloaded on the channel[11]. The basic questions to be addressed are : Which pages are to be put in which queue? In what order will each queue be processed? How will be bandwidth be split between the two queues?
Below, we list some of the possible scheduling algorithms which the MSS can follow:

- **Exclusive On-demand**

  All the pages go into the on-demand queue and the scheduling is by the 1/M/M queuing model. The queuing model can be varied.

  This method can only handle up to a certain number of queries per unit of time, beyond which the system will simply be blocked and the access time becomes infinite. Publishing mode avoids this blocking phenomenon completely. The power efficiency - *queries per Watt* is poor - any request requires a power consuming uplink request.

  This solution is desirable in case energy efficiency is not an issue and the volume of queries is not very high. For example, if the mobile terminals are powered from car batteries.

  Exclusive On demand mode is currently used in WWW protocols in the Internet.

- **Batching**

  The on-demand requests are not handled immediately even if the channel is available. Instead, each request is delayed by L slots before it is downloaded. In this way, other requests for the same page which come during these L slots can be served by the same single download.

  The packet efficiency (queries per Hz) is better than in case of the exclusive on demand model. The power efficiency, however, remains the same as in the previous method. The recommendations are similar as in the previous case.

- **Exclusive Publication**

  All the pages go into the publication queue and the scheduling is simple - the pages are broadcasted one after another, periodically. All the pages are broadcasted equal number of times.

  For small number of queries this methods has very low packet efficiency. However, no matter what the number of queries is, there is no blocking at all. If the directory is provided then this method has the best *queries per Watt*, since uplink transmissions are totally avoided.

  Wireless broadcasting services such as radio and TV use this publishing mode.

  This mode is useful when dealing with one way communication, for instance it is a good solution for one way pagers and pager based communication. As indicated before, this mode is highly useful for dissemination of data "hotspots". Another possible use is in situations when uplink transmission is possible but requires high energy due to the

[11] Note that two subchannels which we use in this paper can be easily modeled by one channel with frames (time slots) preassigned to individual subchannels.

long distance between the mobile terminal and the base station.

- **Frequency Based Exclusive Publication**

  The pages belonging to the publication queue are broadcasted in the ratio of their frequencies.

  This has a better packet efficiency than *Exclusive Publication* and still as good power efficiency *queries per Watt*. Recommendations are similar as in the previous mode.

The next two strategies mix publishing and on demand modes and strikes a compromise between the best queries per Hz and best queries per Watt.

- **Greedy Publication**

  The on-demand traffic has a higher priority than "publishing" traffic and consequently, the publishing frames are downloaded by the MSS only when there is no on-demand traffic. Multicast addresses are still used by the MSS to address published pages. The client, before submitting the uplink request for a page *listens* first to the channel, using a multicast address of a requested page. Only if that page is not provided within a specific period L then the client submits the uplink request. Below, we sketch how the decisions as to which frame is to be downloaded next are made.

  Consider an information service with 'n' pages. Let the frequency of demand of each page D_i (per unit time) be denoted by λ_i. The idea is to publish pages during the unused slots between on-demand retrieval of pages by the clients. During the unused time slots, pages have to be published in such a way that the number of uplink transmissions are minimized (in order to conserve power). Let T_i denote the time period for which pages D_i hasn't been sent (either by publishing or by on-demand).

  The algorithm is to publish page D_j such that the following inequality is satisfied :

$$\lambda_j * T_j \; >= \; \lambda_i * T_i \qquad \forall\, i$$

  This protocol uses the bandwidth very efficiently but the blocking problem, characteristic for the pure, demand based, solutions cannot be avoided. Power efficiency is inferior to the power efficiency of pure publishing methods, since uplink transmissions cannot be completely avoided.

- **Adaptive Publication**

  This is basically the algorithm which we have described in the previous section. There are two basic queues: on demand and publication queues. The decision about the bandwidth allocation between these two queues is made on the dynamic basis based on the request patterns (according to the adaptive algorithm). Publication scheduling is used for the publishing group. This protocol has the best power efficiency for a required packet efficiency.

The above list is by no means exhaustive and there is a broad range of other possible solutions for the server to schedule its services between publishing and on demand modes.

Consider figure 2, which shows a hypothetical service near an airport. The root page of the service, will have a number of options. The user can navigate through the tree by selecting the required hyperlinks. The pages with darkened background (like the root page) are published and the rest of the pages are provided on demand. Consider a leaf page of the flight information link (at the root). This page has various information about a particular flight, like the terminal it is expected to depart from, the gates it will be accessible from, the time of arrival etc. Some of these pages could be provided on demand throughout the day, but 2 hours before the time of departure of the flight,

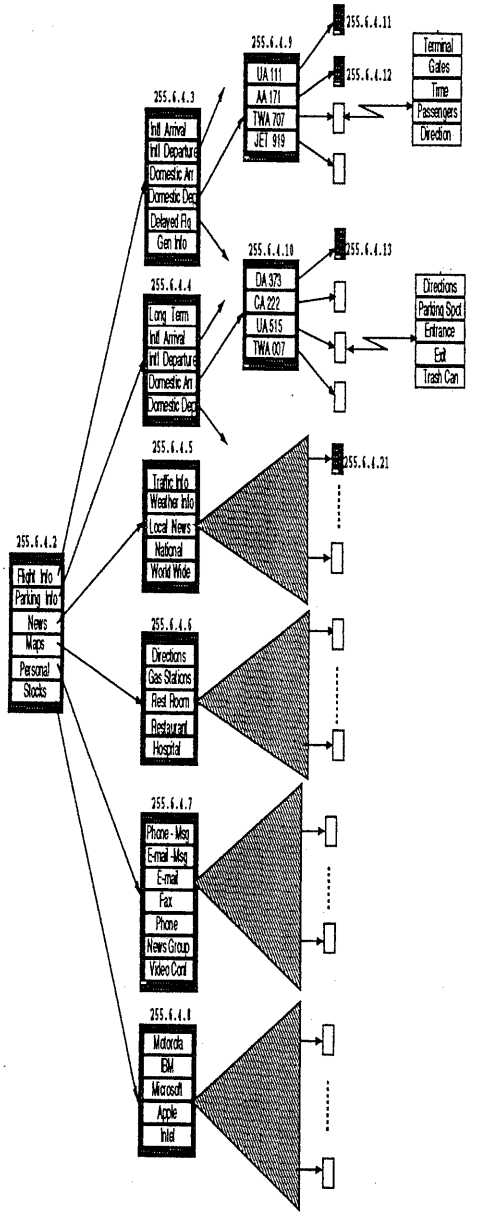

Figure 2: An Information Service near an Airport

they will be published. As the time for departure draws nearer, the frequency of publication of that page will increase and eventually after a few minutes of the departure of the flight, this page will be no longer published and provided on demand again. Thus, the publishing address of a page changes through the duration of the day. Most of the time it is a specific IP address, with which a client can get the page by on-demand mode. A couple of hours before the departure of the flight, the address is a combination of the multicast address and temporal address, the client can get the page by listening to that multicast address say at every 10 th minute of the hour. Just 15 minute before the departure of the flight, the publishing address is just a multicast address which the client can listen and get the required page. Thus, the page has a number of publishing addresses over a period of time.

Note that if the client is familiar with the layout of the information services, it can avoid going through the root and accessing all the pages along the path down to the leaf. For example, if the user knows the flight number and needs to get the flight's arrival time s/he can just specify the flight number followed by the required information.

Influence of Low Powered Buffer Space

Several pagers on the market are equipped with low powered buffer ranging up to 120 Kbytes. In such a case, the client does not have to filter the data "on line" but rather can temporarily store it, and filter it later. Transferring data from the pager's buffer to the computer's memory can be energy consuming especially if it involves standard serial port connection. Thus, the data transfer between the pager and the computer is no longer a low power operation. Consequently, it is very important to provide the pager's stored data with the directory which would help CPU to make a decision whether the buffer's content is to be ignored (and simply overwritten by the new information which comes subsequently). This can be done by providing addressing which is

hierarchical in nature. The first level of the hierarchy would constitute a rough granularity, multicast based addressing. Only packets matching a predefined multicast address would be buffered. The second level of hierarchy would allow further data filtering in order to avoid expensive data transfers between the pager's buffer and the computer's memory.

The second level of filtering can be provided by preceding each buffer sized batch of packets with the directory packet which summarizes the content of the buffer. The CPU, makes the decision about downloading a buffer on the basis of the content of the directory. In this way, the multicast addressing space can be significantly reduced. If B is the size of the buffer, then only every B sized batch of data has to have a separate multicast address. This allows for a fine degree of indexing and consequently finer granularity of low energy information filtering.

For example, a given edition of financial news may be addressed using just one multicast address. The more specific directory can carry the precise description of the financial news content so that the CPU can decide whether the buffer's content matches the predefined set of keywords.

6 Implementing Wireless Information Services

In this section, we will present some ideas on how to extend the currently used protocols such as WWW to implement concepts described in the previous sections. We will also describe possible implementations of such extended WWW protocols on the top of emerging air link protocols such as CDPD.

6.1 Extending WWW Protocols

The proposed family of *Adaptive* World Wide Web protocols will have to support autonomy of local cells in terms of different forms of service delivery, support seamless service continuation upon client's relocation, active and location dependent services, and finally the concept of adaptive links which are available only if predefined resource constraints are satisfied. Thus, in the provided description we go well beyond the set of issues described in the previous sections.

We start by listing some of the disadvantages of the current URL (Universal Resource Locator) protocol used in WWW which make it largely inadequate for mobile and wireless environment:

- No flexibility in terms of the form of delivery of a service: In particular no support for the publishing mode of selected pages.

- Fixed page addressing.

  In URL each page has a fixed IP address. Each client has to make an uplink TCP/IP connection to that address in order to access this page. Thus, there is only one address and one access mode ("on-demand") which is currently supported. This is inadequate if the same page is to be provided "on-demand" in one cell, while periodically multicasted with specific multicast address in another cell. Additionally, the multicast addresses and the periodicity of the multicast may vary from cell to cell. Consequently, there is a need for a dynamic addressing schemes in which the address of a page can vary from cell to cell.

  Finally, indexing of varying levels of granularity may be used in different cells; in this way a page which is a "leaf" in one cell may actually have children in another. Thus, certain pages may simply be unavailable from certain locations.

- No distinction between the *id* of a page and its address.

  This is a consequence of fixed addressing scheme. Consequently, if the same page has two different addresses in different cells it will be treated as different pages. This

is unacceptable in the situations when we want to show the same page with possibly location dependent content to the the client moving across cells. For example a *traffic page* may have different content in different cells and we would like to simply "refresh" it on the users' screen while s/he moves across the cell borders.

Here, we have a loose analogy between IP support of mobile and static users. Mobile users may reside at different IP addresses and protocols like MobileIP have to provide a binding between the IP address of the mobile user and his current location. Similarly, our pages become "mobile" with the same page being bound to different addresses at different locations. This is why we need a page *id*, since the page address can no longer play a role of an identity.

- Inability to specify active services: the WWW servers are *stateless* that is they do not keep any information related to the state of their clients.

- No ability to specify, adaptive links (depending e.g. on the current load on the network)

  Each service can have different presentations which depend on the load of the network and on the client's platform. For example, links with certain broadband type of information such as multimedia (video, audio) may only be provided to clients which are sufficiently powerful and only if there is enough wireless bandwidth on the radio link. Consequently, authoring tools for the future WWW should offer capabilities of *adaptive* specifications which will depend on the available resources: each link should have *resource prerequisites* associated with it and only links which are "eligible" for the current environment (that is a client plus the network) should be displayed on the client's machine. In this way different clients may see different *views* of the same service. In fact, even the same client may see different views of the same service at different times in the same cell due to the varying load on the network.

- Lack of Mobility Support

  Currently, WWW does not support mobility. If a client is logged on in one cell, accessed a page and moved into another cell, then such a client has to restart the service. Despite of having been using a particular page of WWW in its previous cell, in the new cell the client has to start at the root page. One simple solution is to maintain the state of the client in its previous cell at the MSS (of the previous cell) and transfer this state information, as a part of the handoff protocol, to the MSS of the new cell. This way, the client can take off from where it left off in the previous cell. If MobileIP is supported then this feature can be taken care of at the network level. This solution is fine if the page's content was not location dependent. If the page is location dependent then the new MSS has to do some extra work. As the client enters the new cell and registers itself by completing the handoff protocol, the new MSS can take into account that the state information of the client contains a location dependent page of an information service. The new MSS then automatically "binds" the relevant page for the current location and then lets the client start off from that point. This whole process is transparent to the client.

The proposed *adaptive* WWW protocol, should provide all the missing features listed above, along the following directions:

- Different Page Delivery Methods represented by the *generalized page address*

  - On Demand at a specific IP address
    In this case, the IP address is the page's address.

- Periodic Multicast at a specific multicast address and possibly with specific time and periodicity of delivery

  This mode of delivery is appropriate for very frequently requested information. It saves bandwidth and client's energy by avoiding battery draining uplink transmissions.

  The combination of the multicast address and the periodicity of its delivery becomes the page address.

The client will be capable of recognizing whether a given page has to be explicitly requested by making uplink request or it can simply be passively received at a given multicast address and possibly at a specific time without making an uplink request.

- Pages will have page identifiers

  The page address will no longer serve as a page identifier. In this way the "same" page can be linked to different addresses in different cells. It may also have different content depending on the location of the MSS.

- Stateless and Stateful Pages

  Stateful pages will be associated with active services and any service where the requesting client's *ids* have to be stored. If the uplink request from the client is made for a particular service than the id of the requesting client will be stored by the server which is in charge of that page.

  For a popular service, the server may decide to multicast rather than provide a unicast delivery. For example a new high for a popular stock may be announced by multicasting on a predefined multicast address rather than provided on a one to one basis. Similarly, a delay of the departure of a large plane will be better announced by multicasting than by a sequence of unicasts. In the case when multicasting is used there is no need for storing the state information about requesting clients.

  State information may also have to maintained occasionally on behalf of the clients who are disconnected. Consider a Parking Lot picoservice, if a user parked his/her car and leaves, on re-entering the picocell, the user has to be directed to his/her car. This requires maintaining the state information about the users who have parked their car in that picocell.

- Service Hand Off

  If the user is viewing a particular page, upon crossing a cell border, s/he should be able to continue to view that page. The following cases are to be considered:

  - The same page is provided in both cells with the same content
  - The same page is provided in the new cell but with different content
  - This page is not provided in the new cell

  In the first case, the page is simply cached on the client's machine and the cached page is not *invalidated* by the move. In the second case the page is *invalidated* by the move: the client may perhaps get a message informing him about page content invalidation. In the first two cases if the client wants to follow any further links from that page this would have to be the links in the new cell.

  If the currently viewed page is not provided in the new cell, the user should be informed about its unavailability but perhaps keep the connection to the page via the old cell.

  If the client starts using an information service in a new cell the *root* page is provided to him as a part of the *cell greeting protocol*. In this protocol, the client sends an uplink *greeting* message to the MSS and the MSS responds with the address of the *root* page. The root page

may be periodically multicasted on a given multicast address or may be provided on-demand. In the first case, the client simply tunes to a particular multicast address while in the second case, it makes a TCP/IP connection to the specified IP address (and port).

Finally, we should also mention support for continuous services such audio/video. In this case the service handoff is much more sophisticated and requires maintaining continuity as a quality of service requirement.

- Adaptive Links

  It is important to provide the ability to specify *resource prerequisites* which are necessary to access a particular page. This will be related not only to the local features of the client (such as CPU type, memory, pixel resolution, etc.) but also to the networking environment, current link characteristics etc. Consequently, different *views* of the same service may be provided even to the same client at different times and even more often to the same client in different cells and finally to different clients. The *resource prerequisites* can be provided as a part of the service authoring environment and will include: *bandwidth*, *energy* (if the unit is on batteries) etc.

  In this way the MSS will drop some expensive services if the load on the network is too high. The order in which the services will be temporarily unavailable will be a function of the resource prerequisites.

6.2 Cellular Digital Packet Data (CDPD)

CDPD was developed by a consortium consisting of several leading telecommunication companies[12]. It is intended to provide digital data transmission using *existing* cellular infrastructure and thus is likely to be a strong competitor for the existing outdoor packet radio solutions such as RAM Mobile Data etc.

CDPD piggybacks onto (overlays) the cellular network and hence carriers don't have to license additional frequencies from the FCC. CDPD involves adding gear to cellular network to enable them to carry packetized data along with voice. CDPD is attractive mainly because it allows data to be transmitted over much of the current cellular infrastructure, which is mostly analog. It also allows channel hoping, which makes it possible for data transmission to utilize idle cellular voice channels. Any cellular provider is free to offer CDPD based services and several have announced plans to do so. CDPD is expected to grow rapidly over the next year [16].

In CDPD, packet data is sent on unused voice channels on the cellular network. The CDPD subscriber device segments, encrypts and formats the data from the transmitting device (a palmtop or a laptop). The frames are 138 bytes in length and are sent over 30 kHz channels in the cellular network using a protocol called Digital Sense Multiple Access with Collision Detection (DSMA/CD), which is similar to the CSMA/CD protocol used in ethernet. A CDPD subscriber device is an integrated modem and radio. CDPD provides a raw bandwidth of 19.2 Kbps.

The client of a data service in CDPD obtains a connection on a *logical channel*. The logical channel is mapped into a *physical channel* which can be any cellular channel which is currently not used for voice. Data has lower priority than voice, therefore the logical channel may be reassigned to a new physical channel at any time, in case its previously assigned physical channel was taken by voice connection. Thus, a mapping between logical channels and physical channels (called *hoping table*) can change very frequently. How does the client, which is communicating with a MSS on a particular logical channel, know that it has

[12]McCaw cellular, GTE, and five of the seven regional Bell operating companies : Ameritech, Bell Atlantic Corp, Nynex Corp, Pacific Telesis Group and Southwestern Bell Telephone Co

to "hop" to a new physical channel? CDPD informs such a client with a special message sent down on the "old" physical channel, about what the new assignment of the physical channel is. If the client misses this information it has to sniff (hop) on all physical channels to find the one which is currently mapped to the logical channel which this client uses.

In this section, we discuss a way of implementing the *publication mode* on CDPD. Since there is no way to reserve a particular set of physical channels for publishing, we must provide a robust solution which would avoid such reservation but which would still appear to the client as if a reservation was in fact made. For that to take place, the client should be able to know for each multicast address, what logical channel is being used for this address and how this logical channel is mapped to the physical channel.

The proposed solution generalizes the way cellular calls are handled today. The MSS, before publishing a given data item under a specific multicast address, sends on the *paging channel* [13] a short message which has the same multicast address but contains only the physical channel address (i.e. the pointer) of the channel on which the real data item will be sent. The client, upon receiving such a message, tunes to the proper physical channel where the given data item is downloaded. In case, the temporal addressing is used, the temporal address is provided in the initial short message. If multicast addresses are not used (but only a conjunction of temporal addressing on logical channels) then the initial short message is broadcasted on the paging channel (i.e., instead of specific multicast address it has broadcast address). The *Greedy Publication* algorithm described in section 5.2 is used for scheduling the data items that are to be published.

[13]The paging channel in cellular telephony is used to alert the recipients of the incoming telephone call. All cellular terminals listen all the time to the paging channel.

7 Conclusion

We have provided an architecture for future wireless information services. The architecture will consists of a collection of autonomous cells, each offering information services of widely varying geographic scope. The allocation of bandwidth to various services and the form in which to present them, is at the discretion of the MSS. Two basic forms of delivery of information service are available: publishing mode and on-demand mode. We described, how each MSS makes the decision about which form of service delivery to use and how to structure its publication subchannel by varying the frequency of publication of individual data items depending on the frequency of requests. In this way we improve *packet efficiency* and *power efficiency* of the client server protocols by maximizing the number of *queries per Hz* and *queries per Watt*. By making cells *self-explantory* about their service allocation, all mobile clients can interoperate in this environment. A randomized dynamic algorithm for scheduling data items for optimal publishing has been proposed. Another algorithm for optimal allocation of the network bandwidth between the data items to be published and the data items to be provided on demand has been described.

We identify the short comings of WWW which make it inadequate for mobile and wireless environment and propose steps to rectify the defects. We described how the presented concepts can be implemented by proposing an extension of WWW protocol. We also showed how to implement the proposed protocol along the lines of CDPD.

References

[1] Rafael Alonso and Hank Korth, "Database issues in nomadic computing," MITL Technical Report, December 1992.

[2] Daniel Barbara and T. Imielinski, "Sleepers and Workaholics: Caching Strategies

in Mobile Environment," Proc of ACM-SIGMOD, International Conference on Data Management - Minnessota – 1994.

[3] Brinch Hansen, "Operating system principles," Englewood Cliffs, N.J., Prentice-Hall [1973]

[4] T. F. Bowen et. al., "The DATACYCLE Architecture," Comm. of the ACM, Vol 35, No. 12, December 1992, pp. 71 – 81.

[5] David Cheriton, "Dissemination-Oriented Communication Systems," Stanford University, Tech. Rept. 1992.

[6] David Gifford et. al., "The application of digital broadcast communication to large scale information systems," IEEE Journal on selected areas in communications, Vol 3, No. 3, May 1985, pp.457–467.

[7] David J. Goodman, "Cellular Packet Communications," IEEE Transactions on Communications, Vol. 38, No 8, August 1990.

[8] D. Goodman, S. Grandhi, and R. Vijayan, " Distributed dynamic channel assignment schemes," In Proc. of the 1993 IEEE VTC, May 1993.

[9] Jerry Grubb, "The traveller's dream come true,"IEEE Communications Magazine, November 1991, pp. 48–51.

[10] G. Herman et al., "The Datacycle architecture for very large high throughput database systems," in Proc ACM SIGMOD Conf., 1987, pp 97-103.

[11] T. Imielinski and B. R. Badrinath, "Querying in Highly mobile distributed environments," In the Proceedings of the 18th VLDB, August 1992, pp. 41–52.

[12] T. Imielinski and B. R. Badrinath, "Wireless mobile computing: Challenges in data management," To appear in Comm. of the ACM.

[13] T. Imielinski, S. Viswanathan and B. R. Badrinath, "Power Efficient Filtering of Data on Air," Proc of 4 th Intl Conference on Extending Database Technology, Cambridge – March 94.

[14] T. Imielinski, S. Viswanathan and B. R. Badrinath, "Energy Efficient Indexing on Air," Proc of ACM-SIGMOD, International Conference on Data Management - Minnessota – 1994.

[15] T. Imielinski, S. Viswanathan and B. R. Badrinath, "Data on Air : Organization and Access," Submitted for possible publication in Transactions in Data and Knowledge Engineering.

[16] Johna Till Johnson, "Wireless Data: Welcome to the Enterprise," Data Communications, March 1994, pp. 42–58.

[17] Ken Miller, "Cellular Essentials for Wireless Data Transmission," Data Communications, March 1994

[18] D. Raychaudhuri and N Wilson, "Multimedia personal communication networks (PCN): System Design Issues," Third Winlab workshop on third generation wireless information networks, April 1992, pp. 259-268.

[19] Bob Ryan, "Communications get personal," BYTE, February 1993, pp. 169–176.

[20] Samuel Sheng, Ananth Chandrasekaran, and R. W. Broderson, "A portable multimedia terminal for personal communications," IEEE Communications Magazine, December 1992, pp. 64–75.

[21] Lee, William C. Y., "Mobile Cellular Telecommunications Systems," New York : McGraw-Hill, c1989.