Mobile Wireless Computing: Solutions and Challenges in Data Management[1]

Tomasz Imielinski
B. R. Badrinath
Department. of Computer Science
Rutgers University
New Brunswick, NJ 08903

Abstract

Mobile computing is a new emerging computing paradigm posing many challenging data management problems. We identify these new challenges and investigate their technical significance. New research problems include management of location dependent data, frequent disconnections, structuring distributed algorithms for mobile hosts, wireless data broadcasting, and energy efficient data access.

1 Introduction

The rapidly expanding technology of cellular communications, wireless LAN, and satellite services will make it possible for mobile users to access information anywhere and at anytime. In the near future, tens of millions of users will carry a portable computer often called a personal digital assistant or a personal communicator. Various possibilities are shown in Figure 1. Smaller units will run on AA batteries and may be diskless; larger units will run on Ni-Cd packs. These will be powerful laptop computers with large memories and powerful processors. Regardless of size, all mobile computers will be equipped with a *wireless connection* to information networks. The resulting computing environment, which is often called *mobile* or *nomadic* computing, no longer require a user to maintain a fixed position in the network and enables almost unrestricted user mobility. Mobility and portability will create an entire new class of applications and possibly, new massive markets combining personal computing and consumer electronics.

Many predict that *mail enabled applications* [24] will constitute the core of *mobile* computing. Users carrying personal communicators will be able to receive and send electronic mail from any location, as well as be alerted about certain predefined conditions (such as a plane being late or heavy traffic on the way home) irrespective of time and location. Electronic news services will be delivered and filtered according to individual user profiles. For instance, traffic information or weather reports will be filtered based upon the current position of a user, while stock information will be filtered using the user's portfolio. Electronic mail will link applications running on different machines. Staying connected, regardless of location will also stimulate more *collaborative* forms of computing.

[1]The work presented in this paper is part of a larger project called DataMan which addresses all the above issues. The DataMan (a logical successor to WalkMan and WatchMan) project is a joint research effort with Wireless Information Network Laboratory (WINLAB) at Rutgers University.

Figure 1: Personal Digital Assistants

Another class of horizontal [2] applications will include *information services* such as a local yellow pages possibly extended with online information such as movies currently playing at local theaters or merchandise on sale at the local supermarket. Location will play a significant role in selecting relevant information (closest pizza place, hospital etc). Users carrying personal communicators will be capable of processing simple, form based transactions such as the point of sale (sales with credit card verification), inventory orders etc. Such services will require access to databases from anywhere and at any time.

A number of vertical applications of mobile computing already exist. These are the applications built for niche markets that respond to the very specific needs of a mobile work force. Field technicians who require access to manuals while on a repair assignment, fleet management applications (localization and dispatch of trucks) and tracking packages for express mail providers such as Federal Express, are some of these vertical markets. Mobile wireless units are now being used by

[2] *Horizontal* applications are domain independent, as opposed to the *vertical* applications which are written for a specific application domain

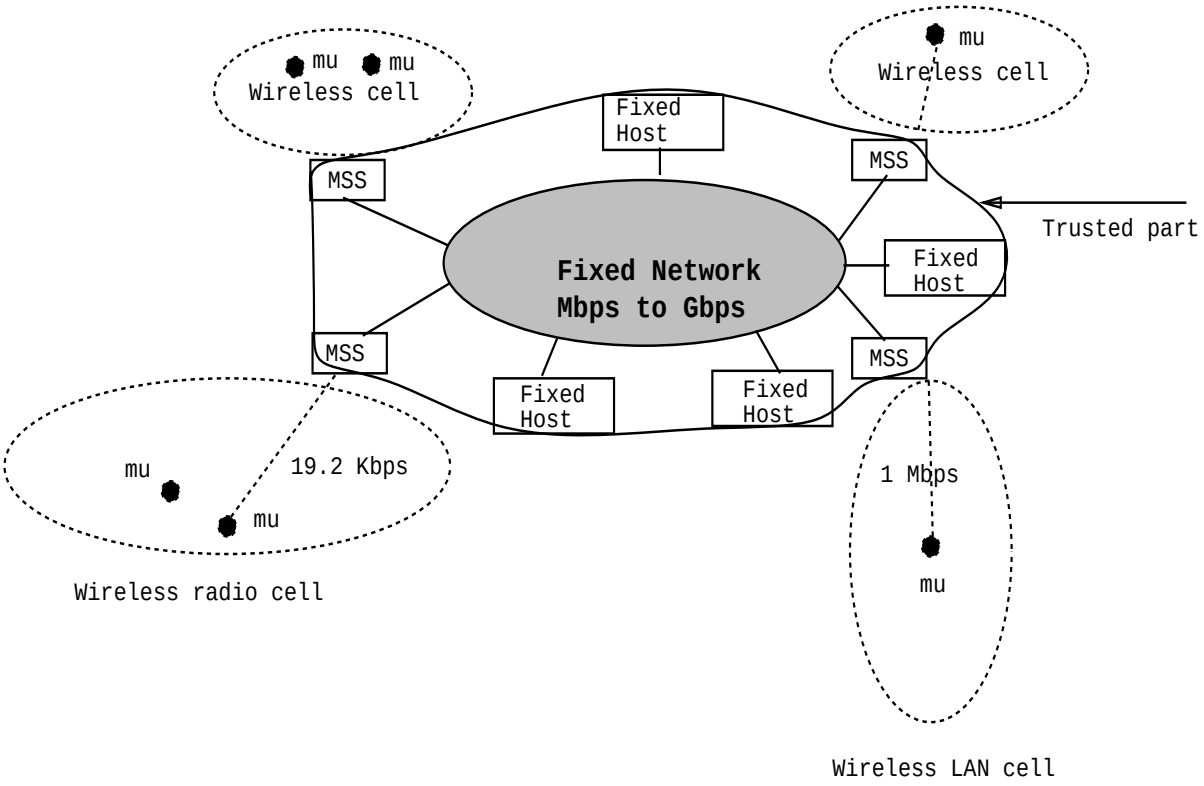

Figure 2: Model of a System to Support Mobility

a number of car rental companies. Yet another existing application of mobile computing is the so called active badge technology, where infrared communication is used for locating employees and redirecting voice mail and data. Hence, needed information can follow the moves of the recipients.

We expect that the *massive market* for mobile computing, which many predict will emerge by the end of the decade, will be based on such mail-enabled and information services oriented applications. These applications will differ from traditional numerical and computationally intensive applications which will typically not be run on small palmtops but rather on the static hosts or, quite possibly on the more powerful laptop computers.

Figure 2, shows our view of the global architecture to support mobile wireless computing. The model consists of two distinct sets of entities: mobile hosts and fixed hosts. We make no assumptions about the types of units that will exist in the system or about the type of data that mobile units will carry. Some of the fixed hosts, called MSS (Mobile Support Stations)[21], are augmented with a wireless interface to communicate with mobile hosts. Additionally, the MSSs will provide commonly used application software, so that a mobile user can download the software from the closest MSS and run it on the palmtop or execute it remotely on the MSS. Thus, the

most commonly used applications will be fully replicated. Since not every file will be carried on the mobile platform, each mobile user will also be associated with a Home MSS, which will store information such as user profile, login file, access rights together with user's private files. Our architecture is intended to support both smaller units which we call *dumb terminals* and larger units which we call *walkstations*. The dumb terminals will rely completely on the MSSs, will have no disk, and usually a very limited amount of RAM. Dumb mobile terminals (sophisticated pagers as well) will participate in any global distributed environment only via "proxies" acting on their behalf and residing on the fixed network. Here the problems are related to *ubiquitous* networking. The walkstations, on the other hand, will do a significant amount of processing locally and only occasionally use the resources of the MSS. Walkstations will have its own disk on which it can cache a portion of the external database, which can be queried and updated. Such a mobile unit can be both a client as well as a mobile server. Even in this model, the issues and problems vary depending upon the kind of data that mobile hosts carry around. The kind of data can be 1) private data (such as, phone numbers, calendar information etc), here data can be cached and updated or retrieved via wireless channel either due to a request or due to a trigger; usually the reader of the data is the sole writer. 2) public data (such as, news, weather, traffic information, flight schedule), where data can be filtered at and queried from the mobile unit. This data is updated by one source and read by several other sources 3) shared data (e.g., replicated or fragmented data of a database, group data etc), where mobile hosts play a role in maintaining the consistency of data and hence participate in distributed decision making and take appropriate actions. This is the most general model where multiple readers and multiple writers exist.

In this model, the MSSs play the role of libraries from which a user can check out books of interest, the palmtop can be compared to a suitcase in which only the most important personal documents are carried around and the home MSS is the user's personal bookshelf.

Fixed hosts and the communication paths between them constitute the static or the fixed network, and can be considered to be the trusted part of the infrastructure[3]. Thus, the general architecture for the network with mobile units is a two tier structure: powerful and reliable fixed network with Mobile Support Stations and a massive number of mobile units connected by slow and often unreliable wireless link.

We believe that the field of mobile computing is at the stage where personal computing was in the late seventies. As we know, research community's participation in the personal computing revolution was minimal, treating (probably correctly) a personal computer merely as a scaled down version of the mainframe. Mobile computing contrary to personal computing has a strong research component. This is due to *mobility* of users. Mobility has consequences for systems designers comparable to that of *distributed* systems. Similar to distributed transaction processing, distributed query processing and distributed recovery, we will have to provide capabilities for mobile transaction processing and query processing as well as recovery for mobile hosts.

[3]The fixed part of the network can be considered to be secure and more reliable than the wireless link and the mobile units

Mobile computing will bring about a new *style* of computing. Due to battery power restrictions, the mobile units will be frequently disconnected (powered off). Most likely, short bursts of activity, like reading and sending e-mail, or querying local databases will be separated by substantial periods of disconnection. Also, quite often, the unit will "wake up" in a totally new environment in some new location far away from home. Finally, due to mobility, the unit may cross the border between two different cells (coverage areas) *while* being active (the so called *handoff* process). To the extent possible, all these changes should appear seamless to the user. In fact, the user should see the same computing environment regardless of his or her current location.

Mobile computing poses new challenges to the data management community. How will mobility of users affect data distribution, query processing and transaction processing? What is the role of wireless medium in distribution of information? How can one query data broadcast over the wireless? What is the influence of limited battery life on data access from a mobile palmtop terminal? How should prolonged periods of disconnection of the mobile machines be handled?

It is useful to group the major challenges brought by the vision of mobile computing into the following categories:

1. Mobility, disconnection and scale

2. New information medium and new resource limitations

All of the above categories are almost completely orthogonal: *Mobility* is a behavior which has effects both within the fixed network as well as the wireless network. How do we find mobile users? How can one partition and distribute information when consumers of this information are mobile?

Disconnection is another important issue. The main distinction between disconnection and failure is its *elective* nature: disconnections can be treated as *planned* failures which can be anticipated and prepared for. There may be various *degrees* of disconnection ranging from a complete connection to a partial or a weak disconnection, e.g., a terminal is weakly connected to the rest of the network via a low bandwidth radio channel.

Scale is another new factor - referring to the massive size of the potential set of users. Issues here cover massive distribution of services and their organization, organization of mediators and information brokers, general questions of knowledge representation and partition of knowledge among different objects in the system.

The wireless medium will provide a powerful new method of disseminating information to a large number of users. New access methods and new data organization paradigms will have to be developed both for providers of broadcast information as well as recipients. Limited bandwidth of the wireless connection and battery power limitations of the mobile hosts are new resource limitations which will substantially affect data management. For example, battery power limitations may lead to new classes of "energy efficient" data access protocols and algorithms.

Data management in mobile computing can be classified as *global* and *local* data management. Global data management deals with network level problems such as locating, addressing, replicating, broadcasting etc. Local data management refers to the end-user level and includes energy efficient data access, management of disconnection and query processing.

These new research problems are consequences of the unique physical characteristics of the computing environment. In particular:

- Small size and weight of the portable terminals

  This contributes to mobility and portability but puts significant restrictions on the human-computer interface due to limitations on the screen size and keyboard.

- Limited bandwidth on the wireless link

  Wireless connection is a decisive factor contributing to mobility. Bandwidth limitations severely restricts the volume of data that can be transferred over the wireless link. At the same time, since the cost of broadcasting over the wireless link does not depend on the number of users, broadcasting is an attractive method of information dissemination.

- Power restrictions on the palmtop platform

  Contribute to disconnections and stimulate energy efficient access methods.

We discuss each of the above mentioned categories by first briefly presenting the work done so far and then describing the major problems and challenges which are yet to be resolved. We will also present some preliminary solutions. We feel that this is an appropriate format for the paper given that mobile computing is still in its infancy.

The paper is organized as follows: Section 2 presents an overview of wireless infrastructures both currently available and planned for the future. In Section 3, we detail the various aspects of mobility, disconnection and scale. Section 4 presents the characteristics of the wireless digital medium and the design of wireless information services. Section 5 summarizes the impact of mobility on standard database issues and Section 6 concludes with a vision of the future.

2 General Architecture

In this Section, we discuss various wireless infrastructures, and next describe various palmtops that are available.

2.1 Networks

The Personal Communication Network (PCN) of the future will provide a wide variety of information services (voice, data, multi-media) to users regardless of their location. The general architecture of such a network is still very much under debate, yet it is clear that it will include and extend existing infrastructures such as:

1. The cellular (in future, microcellular) architecture (analog and digital cellular phones) capable of providing voice and data services to users with hand held phones. The cellular network is connected to the public phone network.

2. The wireless LAN: a traditional LAN (e.g., ethernet) extended with a wireless interface to service small low powered portable terminals capable of wireless access. The wireless LAN is further connected to a more extensive fixed network such as LAN, WAN, Internet, etc.

3. Specialized service oriented architectures such as those providing data broadcasting over unused portions of FM radio or satellite services (paging) for users with special terminals. Some services or applications may use more than one wireless infrastructure (e.g., e-mail using the cellular to the Internet and then to the wireless LAN and vice versa).

The same mobile unit can, in principle, interact with all three different types of wireless networks at different points of time: for example, by moving from inside a building (where it interacts with the wireless LAN) to outside (where it interacts with cellular infrastructure).

In addition, new satellite services are being proposed. Motorola's Iridium, with 66 satellites that orbit the earth at a low-earth-orbit (LEO), is one such proposal. The satellites are interconnected via microwave links, thus forming a global network in space with linkage to the ground via gateways[18]. The gateways are connected to the public phone network and are capable of connecting to other non satellite users. The satellites orbit providing coverage over different areas; a satellite's coverage or cell moves through users rather than users moving through cells in the cellular. Handoffs occur from satellite to satellite. Thus mobile users are provided with instant access to communication and information anywhere in the world including remote areas where no telephone service is available. However, the equipment required to use this communication system is specific to satellite transmission and cannot be used with other wireless infrastructures such as the cellular infrastructure.

The initial applications for satellite systems are predominantly voice and paging. Additional services planned include messaging and fax. Other satellite systems that have been proposed are Qualcomm's Globalstar (48 satellites orbiting the earth), and TRW's Oddyssey, (12 satellites orbiting the earth) which have the same goal of providing information services (mostly voice) around the world.

Below we describe the general communication architecture for PCN. The assumed architecture in this paper, with the understanding that it is still far from being final, is based on the existing structure of cellular telephone networks and also extended towards the so called, 3rd generation cellular networks [28].

System configuration of a cellular network consists of fixed information network extended with wireless network elements. These elements include, wireless terminals, base stations, and switches. The whole geographic area is partitioned into *cells*. Each cell is covered by a *base station*, which is attached to the fixed network and provides a wireless communication link between the mobile users and the rest of the network, as shown in Figure 3. Currently, the average size of a cell is of the order of 1-2 miles in diameter[33]. The need for cells stems from frequency reuse schemes that aim to better utilize the limited radio frequency spectrum available.

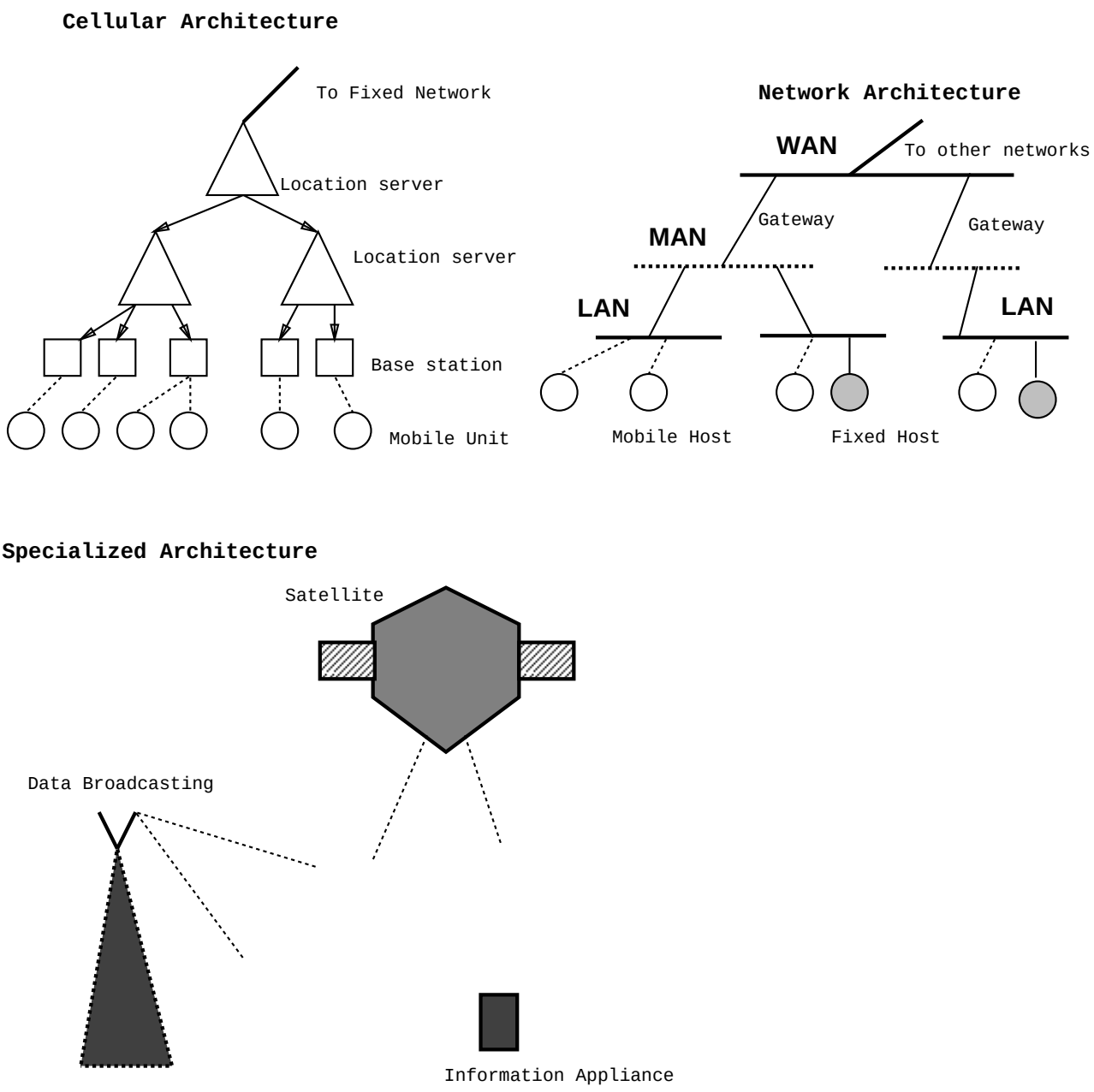

Figure 3: Architectures of Wireless Networks

Additionally, we will assume that there is a hierarchy [4] L of location servers, as shown in Figure 3, which are connected among themselves and to the base stations by the regular network. Typically, location servers correspond to Mobile switching offices and there are about 60-100 base stations "under" a leaf level location server. Higher level location servers cover a larger number of users.

Each user (sometimes called mobile terminal) will be permanently registered under one of the location servers; this location server is also called as the Home Location Server. This association of a user with a particular home location server, is fully replicated across the whole network. Therefore the user's number uniquely identifies the proper home location server. Additionally, a user may also register as a visitor under some other location server. Thus, location servers are responsible for keeping track of the addresses of users who are currently residing in the area "below" the location server. These addresses may be kept as an exact location - i.e., the number of the cell the user is currently in, or as an approximate location such as a zone or a set of cells. Location servers, however, do not have to know in which cell a given mobile terminal is currently located; they can always find out by *paging*, which is multicasting a message to a subset of base stations "under" the given location server. If the call is in progress, as the user moves from one base station to another, a new frequency is assigned at the new base station. The call continues to proceed using this new frequency. This process of transition between two frequencies is called the handoff.

A user's location refers to the address of a cell in which he or she is currently residing. The database that stores a user's current location will typically be located at his home location server but may also be distributed across many levels of location servers as well. This information together with a certain amount of paging is used to contact a mobile user (call set up). This call connection can occur between two mobile users or from a mobile unit to a fixed unit. In either case, the public switched telephone network is part of both call set-up and connection.

Cellular phones mainly provide voice services to mobile users. However, the service is often restricted to metropolitan areas. For example, a cellular user in New York area cannot use the same cellular phone in Los Angeles area. This is because, many of the services available in the regular telephone network are not yet available in the cellular. Examples include wide area coverage, conference calls, and call forwarding to non local areas. More over, the current cellular technology is still analog. Future generations of cellular technology are expected to be digital and are planned to include new services such as multi-media[35].

Wireless LANs are already available. Examples include NCR's wavelan, Motorola's ALTAIR, Proxims Range LAN, and Telesystem's ARLAN. These systems operate in the 902-928 Mhz Industrial, Scientific, and Medical (ISM) band. No FCC license is required to use this part of the spectrum. The data rate specifications of these products range from 250 bps to 2 Mbps with a range of 50 to 100 meters, indoors, and a range of few kilometers, outdoors. The two ends of the wireless link are: 1) a special ethernet hub (or a LAN adaptor) that acts as wireless interface to

[4] Hierarchical structure is modeled upon the general structure of the telephone network. There is no agreement yet, though, on what will be the final structure of the personal communication network

the fixed network and 2) a wireless network interface card (adaptor) attached to the computing unit (workstation, PC, or palmtop). Packet exchange between the ethernet hub and the computing unit takes place via this interface card. A LAN with wireless interface hubs and the interface cards attached to the computing units form the wireless LAN. The wireless LAN can be connected to other networks such as a Metropolitan Network (MAN), a Wide Area Network (WAN) or the internet, as shown in Figure 3.

2.2 Palmtops

Several palmtops are already available in the market. Specifications of some these palmtops are as shown in Table 1. These palmtops fall into two categories. The first category, consists of machines that resemble a data organizer. Examples of these include the machines from Sharp, HP, Memorex etc. The input device on these machines is still a key board. Typical on-board memory ranges from 1/2 to 1M. The processor clock speed range from 5 to 7 Mhz. CPUs operating at these speeds have low power requirements. Hence, battery life on these machines is longer; in the range of 50 to 100 hours. However, this longevity will be significantly reduced, when remote operations are carried out.

The second category of machines have the power of a desktop PC. The typical input devices include a pen as well as a keyboard. Example of these include products from IBM, NCR and Infolio. Memory is typically in the range 2 to 8M. The processors of these machines run at a high speed (clock frequency 15 to 20 Mhz). Minimal power management functions that are included are back-screen lighting switch off and automatic state saving during powerdown. The power sources range from 8 AA batteries to a rechargeable nickel cadmium battery pack. The disadvantage of these higher clock speed palmtops is that of a very low battery life: between 3 and 8 hours of *typical* use. However, this specification of life time of the batteries can vary significantly depending upon the type of usage. External operations such as file transfers and port connections pose a significant drain on the battery.

AT&T, EO corporation, and GO corporation together have announced a personal communicator system. This system has support for cellular voice and data communication. Further, the software platform, the pen-point operating system from GO, is designed for supporting different modes of wireless communication and all types of messaging (fax, e-mail, and voice). Apple's Newton, appropriately called the personal digital assistant, is soon to be released. Newton features a low powered 32-bit RISC processor and the software can be enabled by pen input. Newton offers limited wireless communication based on infrared technology; hence, only line-of-sight communication is possible[36].

In the next two sections we will discuss the main research challenges.

Name	CPU	MHz	RAM	Slots	Battery hours
Abstract R&D	F8680	14	1/2MB	1	50
HP-95LX	NEC V20H	5.37	512KB/1MB	1	50
Memorex	80C88	7.16	640KB	1	–
Poqet PC	80C88	7	512KB	2	100
Sharp PC-3	80C88	10	1/2MB	2	–
Zeos	NEC V30	7.15	1MB	2	10/30
IBM tablet	80386 SX	20	4/6MB	2	3
Infolio	MC68331	16	6 MB	3	12
NCR	80386 SL	25	8 MB	4	3
EO 440	Hobbit chip	20	4 MB	1	-

Name	W x D x H	Weight	Display	Cost
	(inches)	(pounds)		US$
Abstract	8.6 4.3 1.1	1.0	CGA	600
HP-95LX	6.3 3.4 1.0	0.68	40x16	400
Memorex	9.33 4.33 1.16	1.28	CGA	600
Poqet PC	8.8 4.3 1.0	1.2	CGA	800
Sharp PC-3	8.8 4.4 1.0	1.23	CGA	869
Zeos	9.6 4.5 1.0	1.20	CGA	600
IBM tablet	12.25 9.18 1.18	6.1	VGA	3500
Infolio	11.2 9.25 1.25	3.4	VGA	1800
NCR	10 12 1	4	VGA	7000
EO 440	7.1 10.8 1.5	2.2	VGA	2000

Table 1: Specifications for Palmtops

3 Mobility, Disconnection and Scale

Mobility is a behavior with implications for both the fixed as well as the wireless networks. On the fixed network, mobile users can establish a connection from different data ports at different locations. Wireless connection enables virtually unrestricted mobility and connectivity from any location within the radio coverage. In this section, we will argue that mobility is an important new component that will have far reaching consequences for systems design. Purely from a data management perspective, location of a user, due to his mobility becomes a *dynamically changing piece of data* with one writer (the user himself) and possibly many readers. How should the location data be replicated? How should location data be read and stored? We refer to these problems as *location management.*

Further, due to mobility, the system configuration is changing all the time. Consequently, the system resources should be dynamically managed. This includes both the placement of data as well as the agents who manage it (such as transaction coordinators etc). Additionally, all major distributed computing algorithms which rely on a fixed logical structure within the system (such as ring, grid, tree etc.) are seriously affected. We call these issues *configuration management.* Another factor, along with mobility, that will change the global configuration of the system is frequent disconnection (switch off) of mobile terminals to save power. Disconnection will be discussed in Section 3.3.

Below, we will discuss location management and configuration management in more detail.

3.1 Location Management

In the mobile environment, the location of a user can be regarded as a data item whose value changes with every move. Hence, location becomes a *frequently changing piece of data.* Establishing a connection requires knowledge of the location of the party we want to establish a connection with. This implies that locating a person is the same as reading the location "data" of that person. Such a read may involve an extensive search across the network as well as a database look up. Writing the location variable may involve updating the location of the user in the local database as well as in other replicated remote databases (i.e., informing others). Location can also be a subject of more complex aggregate queries. Examples include finding the number of taxi cabs in the stadium area, or looking for the doctor closest to the place of an accident. Thus, location may be treated as a piece of data which is updated and queried. Therefore, location management is a data management problem.

Let us start first with the problem of establishing a location of a user in the mobile network. Suppose that A wants to establish the location of B; should A search the whole network or should A only look at pre-defined locations? Should B inform anybody about his moves? Examples below illustrate some of the possible alternatives.

Example [Go to home location]

This example roughly illustrates how location management is performed in the current cellular architectures[28]. It is also closely related to the way mobility is proposed to be handled over the

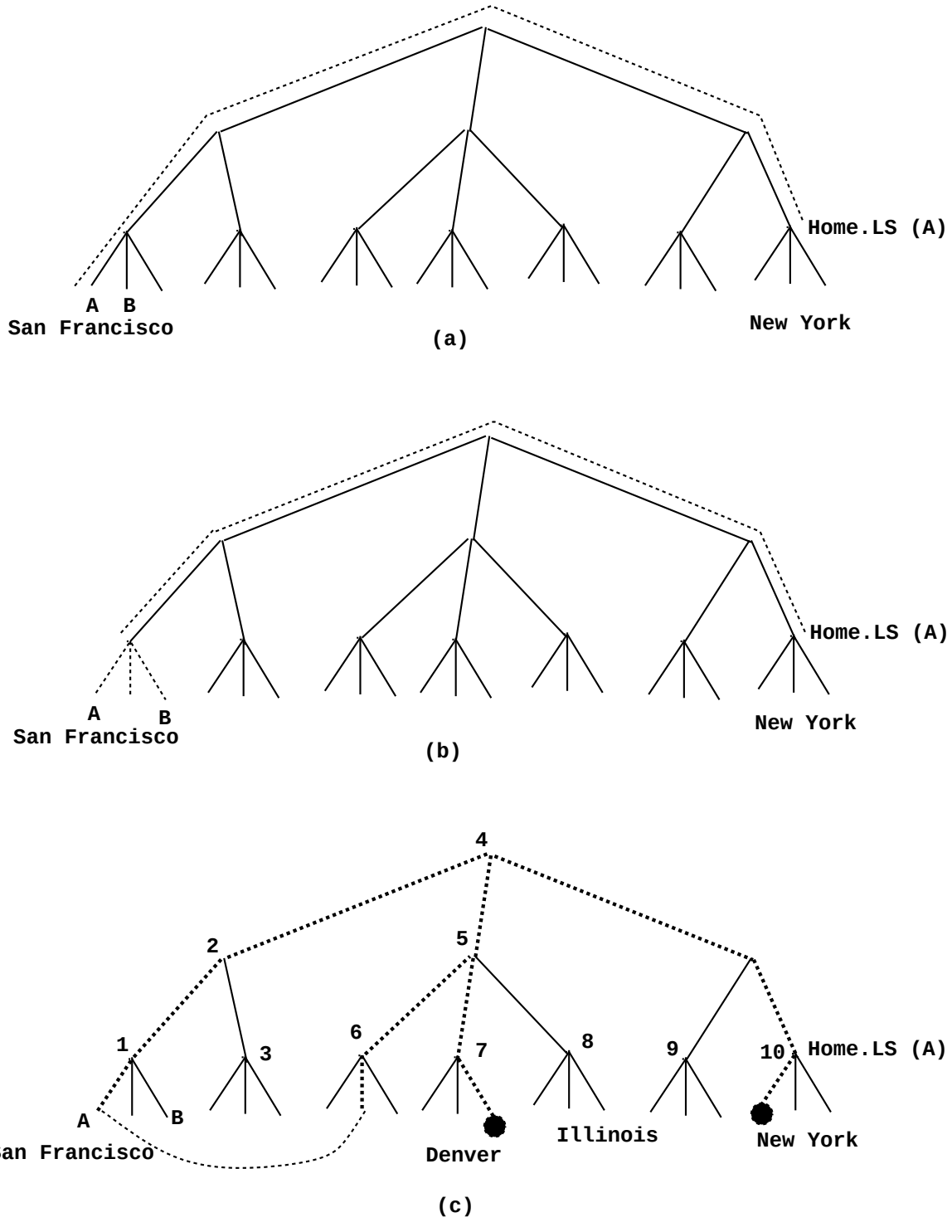

Figure 4: Mobility

Internet [22, 21, 40].

Assume that each user is attached to a home location server which always "knows" his current address. When a user moves, he informs his home location server about his new address. To send a message (or call) such a user, his home location server has to be contacted first to obtain his current address. In [22] a special form of "address embedding" called IPIP is used to redirect the packets addressed to the mobile user from the home location to his current location.

The major disadvantage of this scheme involves the management of so called "global" moves. To see why, let us look at the scenario illustrated in Figure 4 (a):

Consider A, whose permanent registration is in California, and B, whose permanent registration is in New York. Assume that both A and B are currently in California. Suppose that A sends a message to B. In order to route the message appropriately, A has to contact the location server in New York which knows that B is in California. B who may be located just "next door" to A, will receive the message only after two messages are exchanged across the whole country.

This scheme works well for users who stay within their respective home areas (i.e., they never move far away from home). As the scenario above indicates, it does not work well for global moves. In the next example, we show another method which better handles global moves under certain probabilistic assumptions.

Example [Ask around]

This scheme, proposed in [32], is based on the assumption that most messages are exchanged between local parties or between a user (in a remote area) and its home location area (in case he or she is outside of his or her registration area). This scheme is illustrated in Figure 4(b).

In this case, before sending a message to the NY home location server, A, who is located in California, will first, broadcast a message to all base stations in the local area (say within the Bay Area) to find out if B is currently located there. Thus, only if B is not found in A's local area, will B's home location server in New York be accessed. Assuming that this assumption mostly valid , this locating method should be close to optimal. In general, by taking into account, the statistical profiles of user mobility and their calling (messaging) patterns the overall performance of search and locating methods can be dramatically improved. The next two examples illustrate this point more clearly.

Example Calling and Mobility Profiles

Assume that we have at our disposal, for a given user, the spatial probability distribution of his most likely "callers". This is illustrated in Figure 4 (c). Assume that A is most often called from New York and from Colorado (as indicated by darker spots in the figure). In this case, it makes perfect sense for A to inform the Colorado and New York sites about his moves since A's current location will be most likely needed at those sites. Sending a message (or calling) A from New York or Colorado will then be very simple - A's location will be available right there. What if the caller is located in Illinois? There are a number of possibilities: such a caller will have to contact A's home location server (assuming that HLS exists and the address of HLS can be obtained from knowing the user-id of A) or, in case it does not exist, then the caller can perform an *expanding search* from

his current location (Illinois). The expanding search starts from the Illinois area and proceeds to higher levels of location servers and eventually may end up searching the whole network. In Figure 4 (c), the search will start from 8, proceed to its parent, 5, (and all locations below), then, in case of failure, will restart at 4, which effectively means that the whole network will have to be searched.

In all the above cases, the moving user informed some designated locations in the network about *each* of his moves. Such solutions may not scale up well for environments with millions of users in picocell architectures of the personal communication networks of the future. In fact, according to an estimate in [31], the traffic resulting from the location updates in the communication network may exceed the current communication traffic in cellular networks by an order of magnitude. One possible remedy to this problem is not to inform about every single change in location but rather maintain *incomplete information* about the location of the user. But when should we inform? The answer depends on the *mobility profile* of the user:

Example

Figure 5 shows the daily routine of a professor who commutes between the campus and home areas once a day and does a number of local moves within each of the areas. The following options can be considered:

- The network has to page (page is the term used for broadcasting the address of the user in order to find him) the professor across both campus and home areas every time a call is made to the professor.

  The cost would be 7 paging messages per call since there are 7 base stations in the two areas.

- The professor informs the network every time he moves between any two locations.

  The cost would be 8 updates for campus moves (assuming a 8hr stay in the campus area) and 8 updates for home moves (assuming a 16 hr stay in the home area). The cost of a call would be 1 paging message

- The professor informs the network only when he moves from the campus area to home and back.

  The cost would be 2 updates, 1 for each crossing and the call cost is either 3 or 4 paging messages depending on professor's location at the time of the call.

It is clear that the third strategy, assuming the mobility pattern just described, is superior to the other two. In general, any given user could have a *partition* associated with his profile to reduce the overall volume of messages (including paging messages and location updates). Partitions can be defined both at the the global and local level. At the global level, partitions will consist of location servers. At the local level, partitions will consist of base stations. In our example we have assumed that both home and campus areas are located under a common location server. Therefore, the partitions described here are *local*. If, campus and home areas were located under different location servers, then the partition described in the example would be considered *global*.

In [7], we provide a more extensive discussion and experimental results demonstrating usefulness of partitions. In particular, given the mobility pattern of a user, we show how to come up with the

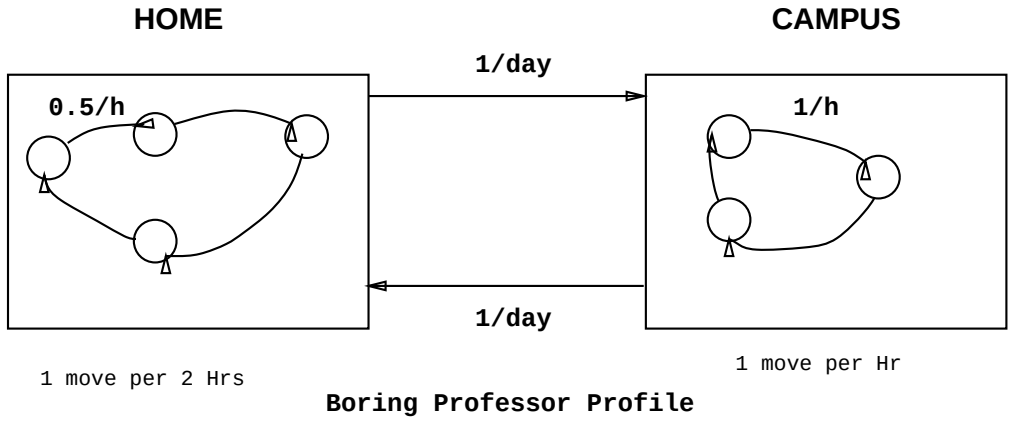

Figure 5: partitioning

"optimal" partition with respect to the overall cost of messages.

Another important characteristic of the user is what we term the *call to mobility* ratio. Call refers to the number of calls made to the user and mobility refers to the number of moves the user makes in a given period of time. Two possible strategies can be applied to locate users: paging (page all the base stations under a location server) or pointer forwarding (contacting a known base station and then following pointers that are updated by the moving user). Which among the two is better depends on the ratio between the number of calls and the number of moves made by the user. For low call to mobility ratios, the paging scheme is beneficial compared to the pointer forwarding scheme. Indeed, since the user is not called often he unnecessarily pays for location updates (pointer updates). The pointer forwarding scheme becomes beneficial for high call to mobility ratios. In such a case, the cost of location updates is "amortized" over the large number of calls - virtually eliminating all search cost in this case.

This short analysis clearly demonstrates that there is no single "optimal" location strategy for all users. For example, depending on the call/mobility ratio, either the paging or pointer forwarding method will be beneficial for a given user. Similarly, different users will be associated with different partitions, leading to different location update strategies. Since mobility is clearly a function of time, more sophisticated schemes will involve time-dependent characteristics of the profiles. One other interesting strategy would be to force mobile users to inform about the location changes only when the move is inconsistent with a pre recorded schedule (i.e., when the user behavior is *deviant!*).

Finally, let us also mention an interesting and general model for locating users developed by Auerbach and Peleg. Auerbach and Peleg in [5] provide a formal model for the tracking of mobile users. Their strategy is based on a hierarchy of regional directories, where each directory is based on a decomposition of the network into regions. The general mechanism is based on intersecting "read" and "write" sets. A vertex v reports about every user it currently hosts, to all vertices in

its pre-defined write set $Write_i(v)$[5]. The searching vertex w, when looking for a particular user queries vertices from a read set $Read_i(w)$. These sets have the property that the read set of the vertex w is guaranteed to intersect the write set of the vertex v whenever the distance between w and v is no more than than 2^i; on an unsuccessful search at a given level, the search proceeds to the next higher level in the hierarchy.

There are two main operations Move and Find. Whenever a user u moves to a new location at distance d away, only the $log(d)$ lowest levels of the hierarchy of directories are updated to point directly to the new address. Directories at the higher levels continue to point to the old location. In order to provide access for users who use those remote directories, a forwarding pointer is left at the old location, thereby directing the search to the new one. Find becomes more involved: nearby searchers must locate u by inspecting their local directories while searchers from remote locations have to use higher level directories which are imprecise and consequently have to use forwarding pointers.

The proposed framework is elegant and can cover most of the locating methods (structure of read and write sets as well as the structure of directories would have to change). It remains to be tested experimentally, how the scheme based on regional directories scheme would perform compared to the schemes that we have explored. Integration of profile information about mobility patterns of individual users within this scheme would be interesting as well.

3.1.1 Queries

Location like any other piece of data item can be a subject of complex queries. such as, "Find the closest doctor to the campus" (where doctors can possibly be mobile) or "Find the number of policemen in the stadium area." Ad-hoc queries like this may be formed by users; they also may be formed by the network or the system administrator to balance the system's load dynamically. For example, the system may decide on the type of message routing or frequency allocation on the basis of individual user locations (as in the former case) or some aggregate information such as a count of the number of policemen in the latter case.

Here, we will discuss only those queries in which mobility plays a role. In general, we will face the following choices for query processing:

- Those that rely only on the database information (that is information stored at the location servers) while processing a query. Since the data in the database may be imprecise (for example, due to partition-based addressing scheme described in the previous section or to the outdated information), the answer to the query will also be imprecise and possibly prone to errors. But the *communication* effort expended will be null.

- Those that send additional messages to find out the exact locations of objects which are relevant to the query. Here, at the expense of additional communication we can find out a more precise (closer to the actual) answer to the query. Obviously, we will be interested in

[5]Subscript i refers to the level of the directory in the hierarchy

Example Consider a query to find the number of taxi cabs in a given area. The database knows only the partitions in which the taxi cabs are located; thus, bounds (between 3 and 6 taxis) can be obtained immediately. These partitions are explicitly represented as numbered rectangles in figure 6. Improving the bounds will require additional messaging. For example, we can find out the exact locations of taxis 5 and 6 and improve the gap between the lower and the upper bounds in the answer by 2. An interesting question is by how much can we improve the bound given that we can use only K additional messages.

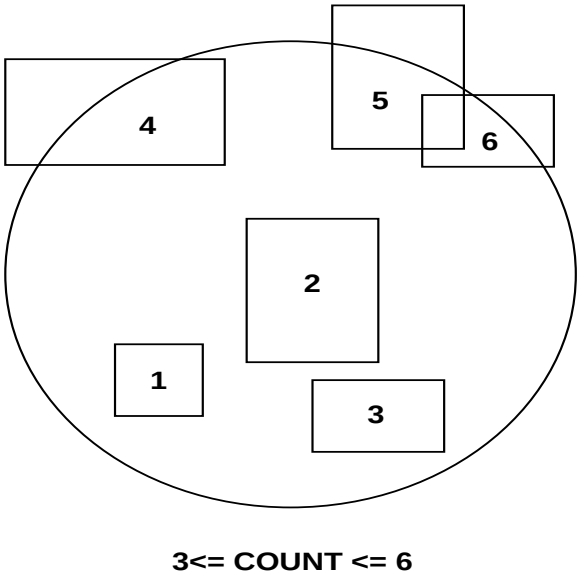

Figure 6: Users in a given area

exchanging as few messages as possible, since communication cost is always substantial.

Hence, we will face a tradeoff between communication cost and the required precision of an answer to a query. Our aim is to achieve "maximal precision" for a fixed communication cost. The precise model is described in [7]. Here we present two examples illustrating the basic tradeoffs between communication and computation. The example in Figure 6, illustrates how approximate answers can be obtained for an aggregate query and the example in Figure 7 illustrates a min/max query.

Querying changing locations requires careful combination of spatial query processing along with the management of incompletely specified data. Incompletely specified location can be "completed" in the run time of a query by additional messaging. Thus, a query optimization problem has an additional new dimension: the cost of acquiring data during the query's run time.

There has been very little work so far on *comparing* different locating and addressing schemes. The problem is difficult, since it involves several dimensions. Solutions which are optimal in terms of numbers of messages sent, may display a very poor performance in terms of latency. Also, it is not clear how detailed the statistical profiles of the users ought to be in order to provide a significant performance advantage. For example, should partitions for the user depend on time? Should we also store some predictions about the typical moves of the user? This information could be quite useful but storing and transmitting it incurs a cost; thus, it is questionable whether or

Example Another example of an aggregate query is the problem of finding the distance to the nearest doctor. The doctors are known to be in partitions 1, 2, or 3. The closest distance can either be MIN if a doctor is found in partition 1 or MAX if a doctor is found in partition 2. The closest distance will lie within these bounds. We can improve these bounds by sending additional messages to determine the exact position of the doctor in partition 1. However, the question is by how much the bounds be improved, given that we can only send K additional messages

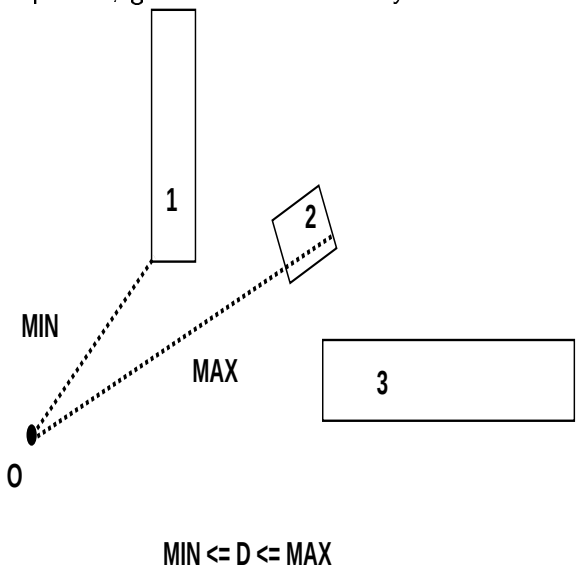

Figure 7: Nearest Object

not the approach of treating users on an individual basis will pay? Yet another problem involves the degree to which the moving user should collaborate with the network in providing information about his position. When should he inform and when should he not inform a change in location to the network? Some of these questions are addressed in [12].

3.2 Static or Mobile? - Configuration Management

Mobility changes the configuration of the system. Mobile clients may find themselves far away from their servers; servers may also move further away from their clients. Thus, system will have to adapt to dynamic reconfiguration of any logical structure within the system and relative distribution of clients and servers. In this section, we will briefly discuss how mobility affects the issues of service placement, where services could range from providing raw data to a complex transaction coordination mechanism. The basic question addressed here is whether the resources should be static or mobile, i.e., follow the moves of the user.

Where to Place Data

Assume that B is the reader of the data item X, which is written (produced) by A. Let us refer A as the *producer* and B the *consumer* of X. If B is mobile, then A has to pay an additional cost if he wants to propagate the latest change of X to B, due to the *search* required to determine the current location of B. In figure 8, the consumer B moves around some fixed location (equipped with

a data server D) and the producer A is located far away and is not informed about the position of B. Where should the copy of the message X be delivered, directly to the consumer A or to the data server D?

Assume that A has to incur a large cost of an expanding search in order to deliver the information to B (since A is not informed about B's moves). In such a situation, the cost of direct delivery to B may be quite high, since A has to pay the huge searching cost every time B relocates from Trenton through Princeton to Patterson.

A better solution to reduce the overall communication cost (i.e. the sum of the producer's and consumer's costs) is to deposit the mailed copy at the data server D and let the consumer pick it from D any time he wants to. In this case, A pays a fixed delivery cost and incurs no search cost while B has to pay the overhead of the communication cost with the data server D in order to read X. If this is unacceptable to B, then he should inform A about his moves in order to reduce A's search overhead. An intermediate option involves sending cache invalidation message directly to B and depositing the copy at D.

As we can see, mobility introduces the cost of search to the global cost analysis. In general, the less informed the party is, the more the search cost incurred. Hence, mobility substantially affects data placement and the mobile consumer may no longer "deserve" a replica of the the data item due to the search overhead which his moves create for the producer. In this case, the data has to follow the user less directly by being replicated either at a fixed location or at the location server currently under; thus, slowly "following" the user. In [8] we provide much more systematic analysis of such replication in the mobile environment.

Where to place coordinators

So far we have discussed a scheme for deciding data placement with a single pair of consumer and producer with relative knowledge about each other's position. Distributed transaction processing may involve large groups of mobile users whose relative positions in the network have to be carefully monitored.

Suppose that a number of mobile users with write privileges have checked out the same data item from the database. One may apply either an optimistic or a pessimistic concurrency control scheme to deal with this situation. But, who is going to coordinate this concurrency control protocol? If the group is mobile, then setting up a fixed coordinator may not be a good idea. Indeed, suppose that the coordinator has been set up in New York while, in fact, the whole group has just relocated to California. It is not reasonable to assume that all members of the group will exchange messages (whether locks or commits) with the remote server in New York. Instead, it is desirable that the coordinator along with the group migrate to California.

Relocation of the coordinator requires much more careful, aggregate location management than which was described in the previous section. One has to be able to constantly track where the "*center of mass*" of a group of users is located, to be able to determine the optimal position of the coordinating site.

In general, mobility of hosts also brings in a new set of issues in distributed systems. First,

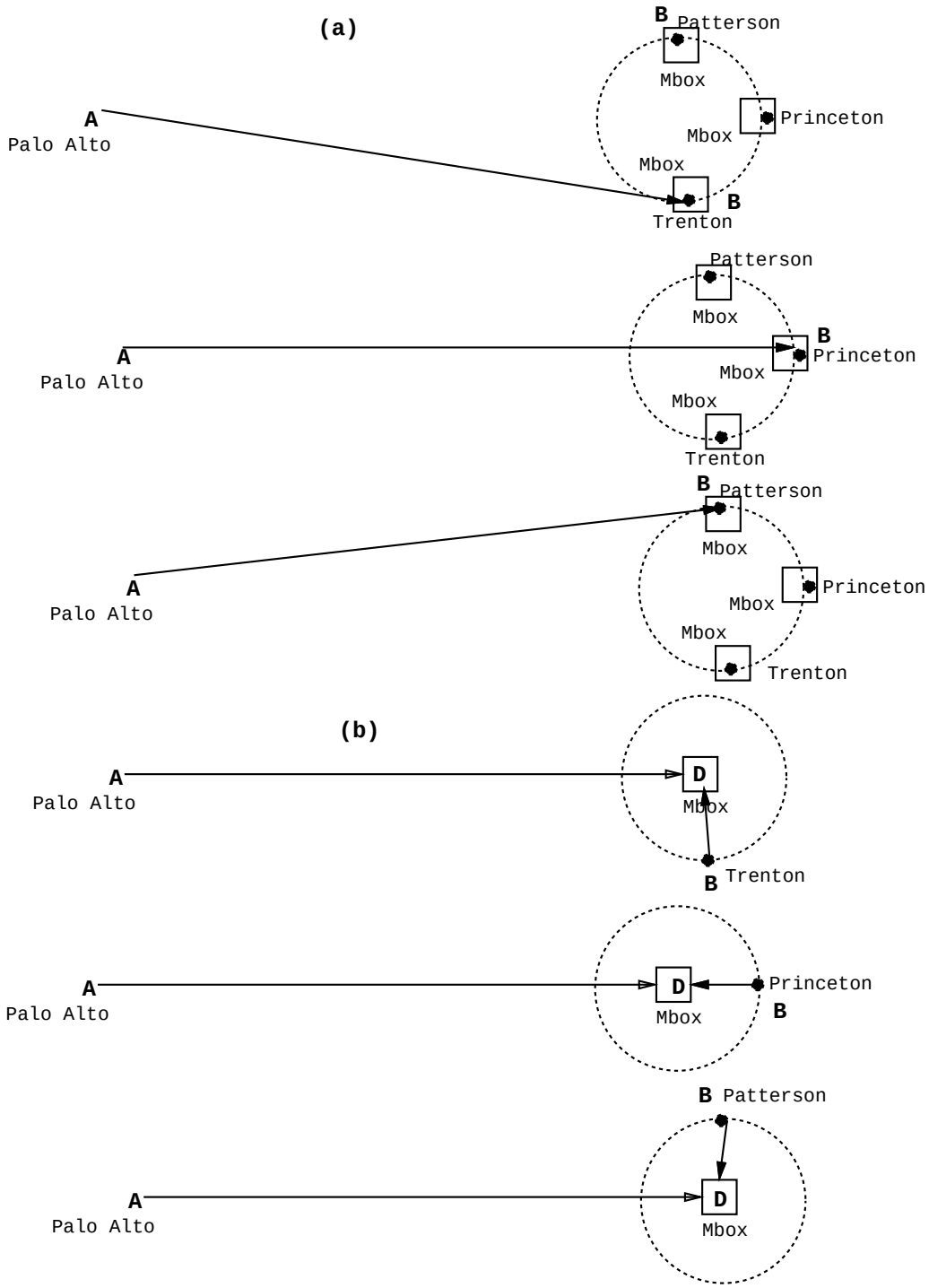

Figure 8: Direct Vs Indirect delivery

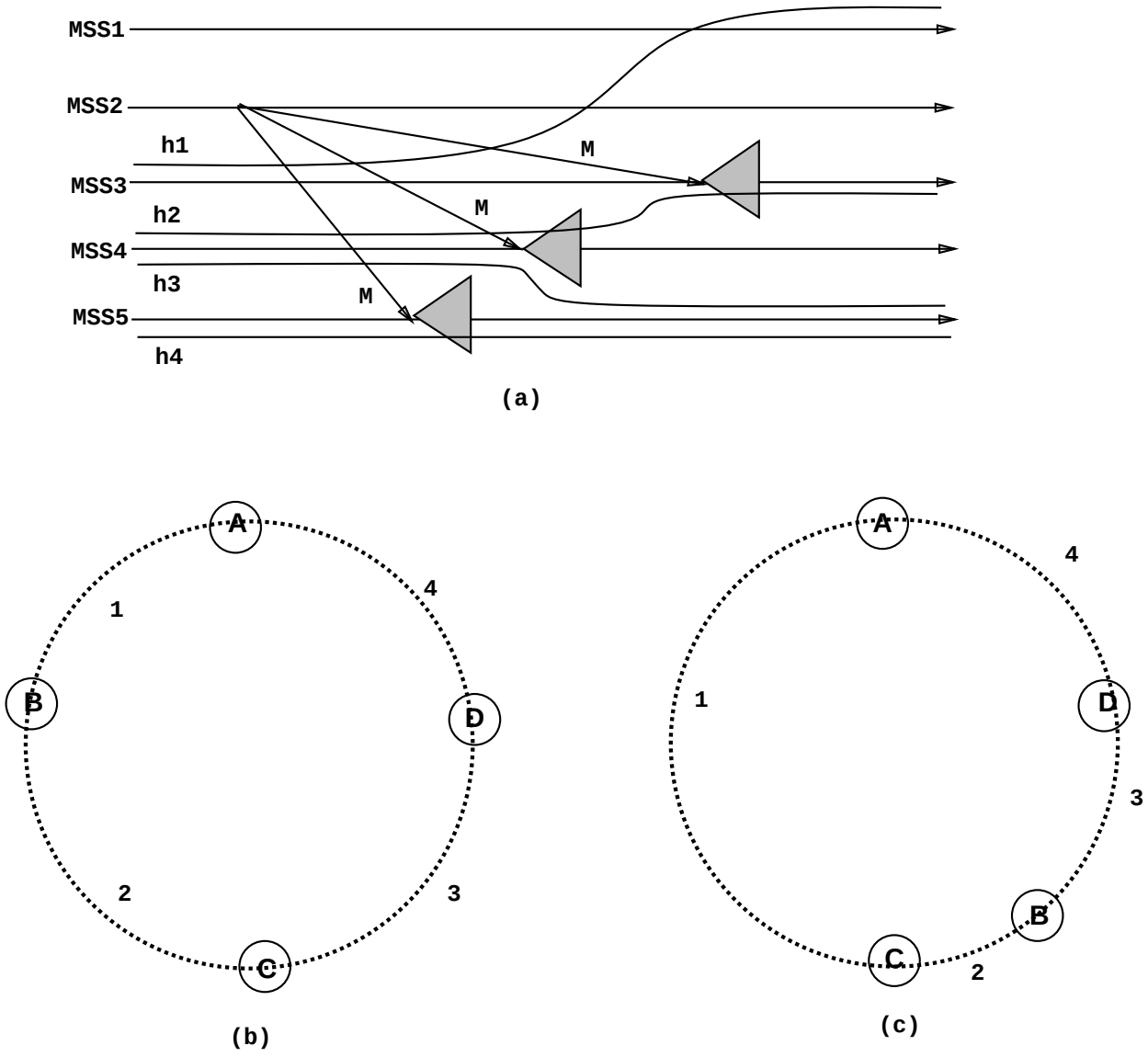

Figure 9: Problems with Mobility in Distributed Systems

the cost of sending a message from one host to another involves *search*, and hence it is no longer fixed for a given <source, destination> pair. Second, as hosts move, physical connections change. Hence, any *logical structure*, which distributed algorithms exploit, cannot be efficiently established among mobile hosts. Along with the severe *resource constraints* in terms of limited battery life and limited size of non volatile storage, mobile hosts have introduced a new set of issues that have not yet been considered in the design of distributed systems.

Consider the problem of delivering a a multicast message "exactly-once" to mobile destinations[1]. Mobility of the recipients introduces two problems: first, two copies of the same message sent over the wired network from a common (fixed) source may reach its destination MSSs at different times due to network latency and second, the same MH may connect to the fixed network from different MSSs at different times and hence, a message may be delivered more than once (In Figure 9 (a), host h_2 receives message at MSS4 and MSS3), or not delivered at all (In Figure 9 (a), h_1 moves from MSS3 to MSS1 thus not getting the message at either MSS3 or MSS1). This problem can be solved if a reliable point-to-point message delivery protocol is available to individually deliver copies to each mobile destination. However, such an approach is not suited for mobile hosts. First, note that since a copy of the message will be sent to many MSSs, the same physical link between two fixed hosts may be traversed many times, each time by a different copy of the same message. Second, in a mobile environment, simulating a multicast by multiple unicasts, one for each intended recipient, may well incur a much higher message overhead due to *search* as compared to a similar scheme for static networks.

Further, as hosts move, their physical connections are reconfigured. Hence, any logical structure such as a ring (see Figure 9 (b) and (c)) cannot be statically mapped onto a fixed set of physical connections among mobile hosts. In Figure 9 (b), the logical connection between A and B is physical link 1. However, if B moves to a new location as in Figure 9 (c), then the logical link needs to be mapped to links 1 and 2. Thus, for every move of the mobile host will incur an additional overhead to reconfigure the set of physical links comprising the logical structure.

A basic approach to designing distributed algorithms for mobile hosts is to localize most of the computational and communication load of the distributed algorithms and protocols on the static portion of the system. The burden on mobile hosts should be minimal due to reliability considerations, to allow for "disconnection" of mobile hosts and for better overall performance. Thus, distributed systems with mobile hosts should be viewed as a two-tier structure: 1) A fixed network with hosts including Mobile Support Stations. MSSs and fixed hosts have more resources in terms of storage, computing and communication and 2) mobile hosts (mh) with limited capabilities connected by a weak connection to the fixed network. For most part, the fixed hosts ensure properties desired of the system, such as mutual exclusion, message delivery, ordering semantics, and consistency of replicated data. Mobile hosts perform minimal additional operations to guarantee the desired overall functionality. Details of this approach are described in [10].

3.3 Disconnection

Another factor, alongside mobility that will change the global configuration of the system is frequent disconnection (switch off) of mobile terminals as a power saving measure. The main distinction between disconnection and failure is its *elective*[6] nature - disconnections can be treated as *planned* failures - which can be anticipated and prepared. There may be various *degrees* of disconnection ranging from total disconnection to weak disconnection. Weak disconnection or narrow connection occurs when a terminal is connected to the rest of the network via low bandwidth (may be intermittent) wireless channel. Let us now look more closely at the features which will have to be supported by various disconnection protocols.

3.3.1 Cache Consistency

If the mobile user has cached a portion of the shared database on his platform he may request different levels of *cache consistency*. While strongly connected to the fixed network, he may want to store, on his cache, the current values of the database items belonging to his cache. On the other hand, a narrow connection may require a more flexible approach of *weak consistency* when the cached item is a quasicopy of the database item. Each type of connection may have a different degree of cache consistency associated with it. Thus, the weaker the connection is, the "weaker" is the level of consistency achieved.

Further, wireless broadcasting seems to be a powerful way of propagating the changes (or cache invalidations) to a massive number of users who are weakly connected to the server through a wireless channel. Broadcast information can either be actual data, invalidations, or even control information such as lock tables or logs. Depending upon the what is broadcasted, appropriate schemes can be developed for maintaining consistency (appropriately defined) of data of a distributed system with mobile clients. Given the rate of updates, the tradeoff is between the periodicity of broadcast and the divergence of the cached copies that can be tolerated. The more the inconsistency tolerated, the less often the updates need to be broadcasted.

3.3.2 Handoff and Recovery

Interesting new aspects arise when the mobile user has to move *during* the execution of a transaction and continue its execution in a new cell or when the user has to disconnect in the middle of a transaction. In the latter case, the partially executed transaction may be left by the mobile user as a *will* to be executed by the local fixed host according to the instructions given by the mobile host before disconnection. For instance, the mobile user may wish to buy AT&T stocks if it reaches a new high today. He may then leave his "will" in the form of an active rule (trigger) at the local host.

Different mechanisms are necessary if the user wants to continue transaction execution after he has reached a new destination. If we assume that a log can be safely stored on the mobile unit,

[6] Term coined by Dan Duchamp of Columbia University

then this does not constitute a major problem. However it is becoming clear that for recovery purposes it is not a good idea to store the log of transaction on the mobile platform. Hence, we will require that the log be stored on the local server. In this case, while preparing to move, the user should insert a record $< IHaveMoved >$ possibly with a destination (if known) into the log. Such a record in the log will be something less than a commit but it will allow the next site to "take it from there". In other words, when the user arrives at the new site, he will continue from that point.

3.4 Scale

The scale of the mobile environment just described will go far beyond any of the existing paradigm. Many predictions call for tens of millions of machines of varying size which can move across a worldwide communication network. Such grandiose scale affects all the issues discussed so far. In location management, the total volume of transactions due to location updates may be so high that the existing network will not be able to handle such high volumes [31]. In configuration management, due to frequent changes which may involve wide moves of large number of machines, scale plays a critical role too. After all, we are talking here about distributed system with not just tens or hundreds of sites. Scale has major consequences on limited bandwidth resources: the increasing number of users requires using smaller and smaller cells. This in turn complicates the location and configuration management due to increasing number of handoffs.

Massive scale of the system also results in its *heterogeneity*. How do we make sure that a mobile unit sees the same or similar environment regardless of its location? How do we provide a uniform access to information services and databases across the network? These issues are certainly not new and are subject of very intense research in the database community already but the environment discussed in this paper makes them even more important and critical. Mobile users can query and *be queried* at the same time, assuming roles of servers and clients. Thus not only servers but also clients may be mobile. How do we find the relevant information in a massive environment like this? Many levels of information brokers and mediators [41] will have to be developed in order to facilitate the information access.

4 New Information Medium, New Resource Restrictions

Wireless broadcasting will provide data "in a new form" that is literally "on the air". Bandwidth limitations as well as battery power limitations will define new cost measures for accessing data and consequently may favor new solutions. We will first discuss the wireless broadcasting as the challenging information dissemination medium.

4.1 To broadcast or not to broadcast - Wireless Information Service

There are several examples of queries which are asked repetitively by a huge number of users in a local area. Local traffic information, stock market data, local sales, events, emergencies, news and

bulletin boards are all examples of information that would rather be broadcasted than provided "on demand" basis. As we pointed out before, the cost of broadcasting over the wireless does not depend on the number of users who are listening. Therefore, the wireless medium is almost an ideal choice for any "public" information service and, in fact, can be viewed as another memory medium (public "air" memory with a latency due to the broadcasting period). Gifford in [17] describes design of Boston Community Information System (BCIS) which uses FM Subsidiary Communications Authority (SCA) channel. SCA channels are subcarriers that can be used by an FM radio station without interfering with normal broadcasting. BCIS broadcasts two wire services (the New York Times and the Associated Press) to test audiences who own personal computers with radio receivers. Each user has a profile which is used by the PC to filter relevant packets of information from the broadcast and then store them on the disk. In this application, the receiver is a relatively powerful PC and the information broadcasted is not accessed in "real time" but is rather downloaded onto the PC and then accessed later. The DataCyle project at Bellcore [19] utilizes broadcasting over a very fast fixed network in order to disseminate a large database to a large number of users. Special hardware allows users to apply on line and in parallel massive numbers of predicates in order to "filter" relevant information from the data stream. Bandwidth, in this case, is a much less significant problem than that of wireless broadcasting. We expect, however, that many of the data access techniques developed for DataCycle will apply to the wireless broadcasting as well.

In a mobile environment, the broadcast becomes an even more powerful medium. Information will be broadcasted to a potentially large set of authorized users (those who have paid for the service and are equipped with the decryption keys). Broadcasting over the wireless medium is also a *power saving* technique from the end-user's point of view, since they do not have to resort to the power consuming uplink transmission but only "listen" to the data broadcasted over the "downlink" channel.

Additionally, since we cannot assume that the palmtop machines will necessarily be equipped with disks we have to reject the concept of downloading the data on the palmtop and querying it later (even if this information is already filtered). What we envision, is rather a main memory based system which will in real time display and subsequently disposal of information received from the broadcast.

Suppose that the local information provider has to his disposal a radio transmitter (base station) which can broadcast information to a cell or to a number of cells. Now, what information should be broadcasted and what should be provided "on demand" and how should the broadcasted information be structured?

- An index "channel" providing a directory to the information broadcasted should be offered in addition to the broadcasted data as indicated in Figure 10. User terminals will listen to the index channel to determine how to selectively tune to the particular information on the "data channel" without continuously listening to the data channel (channels). This is analogous to a TV or Radio program guide which provides a timetable for all the programs and allows us

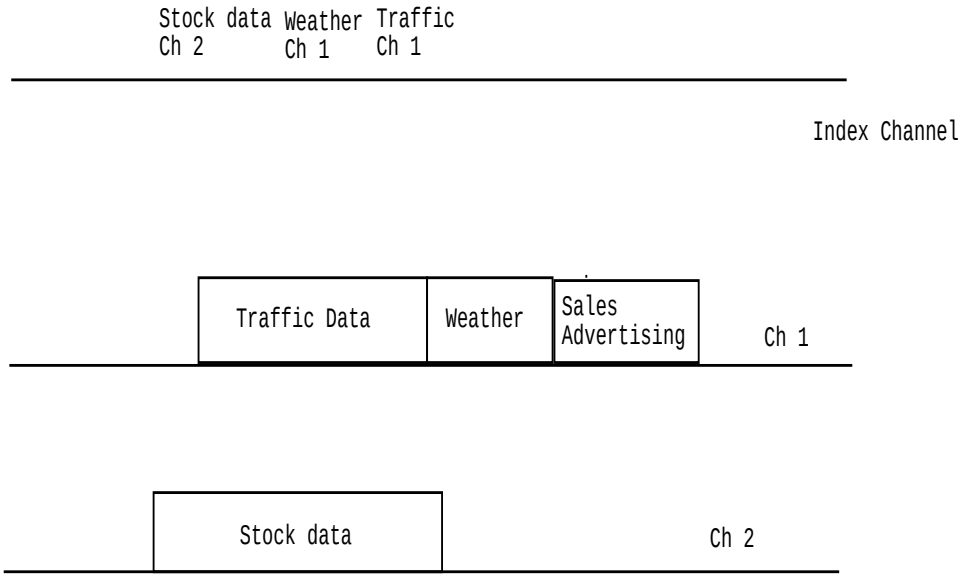

Figure 10: Wireless Broadcast

to watch, listen, or record selectively.

- Analogous to the TV rating system, (Nielsen's rating of the average number of households tuned to a particular TV program) periodic decisions should be made on whether to keep broadcasting a particular piece of information or rather make it available "on demand" basis. Thus, if the demand for the particular a data item goes down it will no longer be broadcasted.

- We could view broadcasting as a purely passive "read only" form of communication. We could also envision more interactive scenarios. For example, after a broadcast of data about available tickets, users may place orders for them. The next broadcast should then reflect, in some way, a new state of the system resulting from such updates.

New protocols will have to be provided in order to implement adaptive broadcasting schedules. Here, we discuss two different protocols which guarantee a latency no larger than L for user requests:

P_1: A base station can be in of two states: waiting and responding. If a query is submitted by the mobile user and the base station is in the waiting state, then the it switches to the responding state and remains in that state for a period L and then broadcasts the answer. If the base station is already in the responding state then no action is taken upon receiving a query (since the system will respond anyhow in at most L seconds).

P_2: The base station broadcasts information every $L' > L$ seconds. The mobile unit, before submitting a query, checks how much time is remaining before the next broadcast. If it is less than L, it does not submit a query request; if it is more than L then a request is submitted.

For the protocol P_2, we assume that an "index channel" exists. This index channel provides the information L', i.e., the interval separating two successive broadcasts.

The periodicity of the broadcast largely depends on the size of the data itself. Indeed, if the broadcasted file is very large and if the item of interest constitutes only a small portion of it, the

user will have to wait in the worst case for a period of time equal to the broadcasting time of the whole file. For example, if it takes 10 seconds to broadcast the local traffic information, then it may take 10 seconds to "capture" the specific information about traffic on Route 1.

The more specific the index information the more precise the information provided to the mobile terminal regarding the timing of the relevant data being made available. On the other hand, a more specific index needs a longer duration to broadcast and additionally contributes to the global latency of broadcasted data. Therefore, the tradeoff between the latency time and the "tuning time" (which is proportional to the time the portable unit has to "listen to a channel") has to be investigated. Intuitively, the smaller is the index, the less precise the information the portable has about the data and the longer time it has to "tune" to the particular channel.

The following example illustrates why minimizing the tuning time is so important:

For example, queries may simply ask for displaying attributes (or methods) of certain objects. For instance, the user may want to display the value of IBM's stock (ibm.stock), the temperature in New Brunswick (New Brunswick.temperature) and finally, the traffic conditions on the New Jersey turnpike (turnpike.traffic). We assume that this information is being broadcasted. Additionally, the user may want to specify *how often* he wants the information to be *refreshed*. For example, he may wish to have traffic reports every 5 minutes or every time an accident takes place; he may want ibm stock 3 times a day and the weather information 2 times a day. On the other hand, the broadcasting service may have its own specification of the broadcast specifying which classes and which methods will be broadcasted, how often and what will be the channel allocation. Some other questions that need to be addressed are:

- How should the index information be structured and how will the index be allocated to the communication channel?
- What will be the specification language that will help the broadcasting service provider specify what parts of the database are to be broadcasted and when?
- How to switch from broadcasting to a point to point (on demand) mode?

In the next section, we discuss another resource limitation which may have a profound influence on the way data access algorithms are designed on the mobile platform.

4.2 Energy Efficient Data Management

The lifetime of a battery is expected to increase only 20% over the next 10 years [26]. This is a extremely slow progress comparing to the dramatic improvements in the CPU speed and memory capacity. Typically 1 AA cell is rated to give 800mA-Hr at 1.2 V (.96 W-hr). A Ni-Cd battery pack is rated at to give 2.4 A-Hr at 10.8V (25.92 W-Hr).

The constraint of limited available energy will drive all solutions to mobile computing on palmtops. Let us assume 10 AA cells (9.6 W-hr) as the power source to a palmtop with a CD-ROM and a display. The constant power dissipation in a CD ROM (disk spinning alone) is about 1 W.

A typical display consumes 2 to 3W (average of 2.5 W). Assuming nothing else is powered on, the batteries will only last for 2.7 Hr (9.6 W-hr/3.5 W). Thus, to increase longevity of the batteries, CD ROM and the display may have to be powered off most of the time.

Power is also consumed by other units of the palmtops such as CPU and memory. The power consumed by the CPU is proportional to the clock speed. Measured power consumption on a 386 laptop is about 0.2 W per Mhz. The 386 processor with 20 Mhz clock consumes 4W and the low-power version 386SL consumes 2.16 W. Hence, the need to slow down the CPU or the clock rate while a palmtop is operating on batteries. However, the new low voltage CPUs have significantly lower demands on power. For example, the Hobbit chip from AT&T operates at 20 Mhz, 3.3V, and consumes 250 mW. The power consumed in doze mode is only 50 microW.

In case of DRAM, the power consumption for a memory bank of 4M is in the order of 0.5W. Even with flash memory, the power required for access is 20 times more than when the flash memory unit is idle. Finally, transmitting and receiving data can consume additional power as well. The power consumed while transmitting depends upon the range of transmission. In general, transmitting a given amount of data consumes twice as much power as receiving the same amount of data. From the point of energy consumption, the decision is between local (palmtop) computation and remote (server) computation - low powered units will only receive data and most of the computation will be done at the server. However, the downside of this mode of operation is the constant power dissipated while waiting for the server to respond [2].

Power brings an interesting new cost aspect to data management during weak connection. For example, transmission (and also, to a lesser degree receiving) should be done when closer to the radio server. Hence, one can visualize a situation when the client knowing the approximate power cost of transmission of messages from the place he is currently located, decides to move closer to the server in order to transmit or download the information from the wireless broadcast. Other issues involve:

- Tradeoffs between memory and channel access: Memory consumes power but so do channel accesses, in particular transmissions. Given a computation is it better to execute on the mobile client or on the fixed server from the point of view of power consumption? How should data be partitioned between the client and the server?

- The role of data compression as a power saving tool. Compressed data uses less memory and communication channel but takes additional CPU cycles (and hence palmtop power) to decompress. It is twice as expensive to decompress than it is to compress. What are the tradeoffs here?

- The role of data broadcast, as a "listen only" data access mode, as a power saving measure. Listening to the broadcasted data saves power by not having to transmit requests for data. However, the receiver needs to be "active" to listen to the broadcast. If the receiver is smart it can be tuned to become active at the appropriate time to receive the broadcast. How should protocols to support this kind of data access be designed?

5 What is affected

A natural question to ask after reading the above sections is "how will all of these new issues affect my work?" So far we have specified new issues brought about by the mobile wireless computing environment. How do they affect traditional data management research areas? Below, we present briefly how specific areas will be affected. In general, we believe that mobility and portability may have as wide ranging an impact on systems design as distribution had in the past. Distribution affected transaction processing, query processing, crash recovery, physical data structures including data placement, security and integrity and many other general systems issues such as operating system design. On the other hand, distribution did not have a profound impact on the query language design, logical database design, or data modeling. We believe that moving from static to mobile environment will have a similar scope of influence as migrating from centralized to distributed systems. Thus, instead of asking "centralized or distributed?" we will simply ask "static or mobile?"

Most of the material presented below has been already discussed - this role of this section is simply to summarize and categorize the information which is otherwise spread across the whole text. We have created three categories to reflect the impact of mobility on various aspects of data management: large impact, some impact, and little or no impact.

5.1 Large Impact

Distributed Data Management

Techniques for distributed data management have been based on the assumption that the location of hosts in the distributed system does not change and the connections among hosts do not change. However, in mobile computing, these assumptions are no longer valid. Hence, distributed data management will be fundamentally changed due to mobility.

Data Management

Maintaining data on mobile hosts implies additional search cost to access the data, consumption of scarce energy resources, and even unavailability of data due to frequent disconnection. To over come these drawbacks, the approach to designing algorithms to manage distributed mobile data should be, to the extent possible, to shift the the computation and communication costs of an algorithm to the static portion of the network; the number of operations performed at the mobile hosts and thereby, the power consumption is kept to a minimum.

Dynamic Configuration Management

Assuming that data, control, and services for most part are maintained on the fixed network, mobility of users will affect issues such as data placement and coordinators placement addressed in Section 3.2. When mobile hosts move to a new area, a new server or a coordinator may have to be added. Thus, the set of servers may change during the execution of a protocol. This new coordinator will then have to interact with other servers to efficiently accommodate this change. Note that a change in the configuration of the fixed hosts providing a service is not brought about by a failure and would be inefficient to treat it as such.

We have already compared the expected impact of mobility to the impact of distribution. Mobility and distribution can be viewed as new features of the physical environment; they can also be treated as *systems solutions*. Mobility of system's resources is a solution for a mobile environment just as distributed processing is for a distributed environment.

Query Processing

Querying Fast Changing, Update Intensive Data

Location is a frequently changing data item. How do we represent and query fast changing, *update intensive* data like this? Precise tracking of all updates may be impossible or simply unnecessary. Instead, we may have to store incompletely specified information. Consequently, queries will be answered in an *approximate* way. The quality of the answer will depend on the computational and communication resources that we are willing to spend. Given the error which we are willing to tolerate, what is the optimal query evaluation strategy? We believe that querying fast changing data leads to a new class of query optimization problems.

Querying wireless, broadcasted, data

Another new problem involves querying data broadcasted by the server and addressed to a large number of clients. On the client's side: what is the best execution plan for a query which involves data broadcasted on different channels? On the server's side: what should be the organization of the broadcasted data (index channel etc?), which information should be broadcasted and which should be provided "on demand"? How will *contiguous* [7] queries [38], be evaluated using broadcasted data?

Power

Finally, as indicated in [2] energy efficient query optimization techniques will have to be developed in order to deal with the battery power limitation problem. Battery power provides a limited resource on the palmtop's size. What are the query execution plans which minimize the battery power consumption? Queries can be executed by entirely or partially by the remote server, which saves energy consumption on client's CPU but increases power consumption due to data transmission. Energy consumption leads to a new class of query optimization problems.

Security and Integrity

Security and privacy is a major controversial issue in mobile computing. Presently, it is largely ignored which is typical for a discipline in its infancy. Here we will concentrate on the software issues and will not discuss the very important and difficult problem of security of the physical wireless channel.

Profiles

Being reachable at any location and at any time creates great concern about privacy issues among the potential users (see for example [16] for impressions after early developments of active badge technology). To protect privacy it is necessary to develop sophisticated software for specification and enforcement of personal profiles of the users. In such profiles users should be able to specify who, when and where is authorized to reach them (receiving e-mail consumes battery power

[7] Queries are contiguous if we want to keep track of the value of the query in a changing environment all the time

of the local terminal; so one may want to restrict the list of users who are allowed to "wake up" the mobile unit to send a message to it). Management of profiles is a nontrivial issue - where are such personal profiles are to be stored? Should the profile migrate with the user or follow him?

Trust

Mobile users will use resources, including software, at various locations. These resources may be provided by different service providers. Thus, the concept of trust needs to be developed to allow mobile clients to use resources of different servers at different locations [3].

Integrity

Enforcing integrity constraints on data which is stored on the mobile platform is a new and a highly nontrivial issue. Consider, for instance referential integrity between two relations each of which is stored on a different mobile platform. Upon each update to the dependent table, the parent table has to be located possibly involving expensive search for its owner. If this is too expensive, perhaps the owner of the parent table should be obligated to inform about his position so the amount of search is cut down? Other problems include maintaining *consistency* of information about location. Since this information is almost never up to date new methods for maintaining *quasi* consistency have to be developed. For instance, similar to [6], we may only guarantee the correctness of the location information within a certain partition (i.e. we may guarantee that X is on the campus, we cannot guarantee that he is still in his office). Data placement is another issue related to integrity. Perhaps one should place the data which is involved in a global integrity constraint on the static rather than mobile platform? In general, *Distributed integrity checking* is greatly affected the mobility of the participants. Given a constraint which involves a number of mobile hosts, how can it be efficiently maintained. Who should inform whom about the location changes and where should the predicates be evaluated? On the static or on the mobile hosts? Should triggering or polling be used when verifying such distributed constraints. This, in particular, involves *active rules* like "let me know if Joe is on the campus" or "if Joe is close to the police car" which critically depend on location. How should the rules of this form be evaluated to minimize the number of messages.

Interface Design

Data Entry Limitations

New interfaces which do not rely on keyboard have to be developed in order to deal with the physical limitations of the mobile terminals. Pen based and speech based front ends are leading contenders.

Data Output Limitations

Due to the limitations on the size of the screen, one has to look for new, more innovative, data output methods for query answers.

5.2 Some impact

Transaction Processing

Form Based Transactions

Transactions submitted from mobile terminals will have much simpler, *form based* structure than the general transactions which are general purpose programs with embedded database calls. These forms can be broadcasted to a number of users, and the users will then respond by entering specific data values.

Long Disconnections

Since the mobile terminals will often be disconnected from the fixed part of the network transactions will often be processed locally on the *cached* data. The degree of "connectivity" of the mobile terminals to the fixed part of the network will vary widely. Mobile users will "check out" portions of the database for long periods of time. Thus, new methods of cache synchronization reflecting different degrees of connectivity will be necessary.

Locality Principle for Concurrency Control

The same data item may be cached on the mobile terminals which are far apart. Even if such terminals are connected to the network, the use of pessimistic concurrency control is questionable due to the large communication cost. We believe that in most applications, mobile users will perform transactions over data that has to do with the local area. Data satisfies the *locality principle* if the most likely access conflicts occur between users who are located close to each other. In such cases, locking will be used only in an area that can be considered geographically local, while optimistic concurrency control methods will be used on "nonlocal" data. Thus, any data item can either be pessimistic (required locking) or optimistic (no locking) depending on location, time of the request and the identity of the requesting user.

In general, basic concepts such as *locking* and *commit protocols* have to be redefined in the context of mobile hosts carrying shared data.

Recovery

New problems here arise to various forms of *elective* failures - failures which can be anticipated and prepared such as handoff and various forms of disconnection.

Disconnection and Reconnection

If the mobile host is taking a part in some process which involves other parties on the network the log of such process has to be stored before disconnection at some static local host. Upon reconnection, the mobile host will contact his local "proxy" in order to *refresh* its state information about the external world. Both disconnection and reconnection require new protocols and modifications to the standard recovery process.

Handoff

When a mobile host moves across different cells while involved in some process we face disconnection followed by reconnection at a different static host. This requires new methods for transfer of state information between the static hosts in order to reconstruct the current state of the process at the current location of the mobile host. Handoffs should be as *seamless* as possible and should enable users to see the same environment "following" them, while they are on the move.

In general, we expect that durability of data cannot be guaranteed with the standard logging techniques. The disk of a mobile unit cannot be considered completely stable, due to the possibility

of physically dropping the unit or loosing the unit totally. Hence, logging should involve the trusted part of the overall infrastructure[3].

5.3 Little or No Impact

Data Modeling

New techniques will be necessary for user's profile specification. These may involve temporal, spatial and probabilistic rules to describe the mobility pattern of the user, and specifying access rights based on location and identity of both the subject and the object.

We see little impact of mobile computing on the data manipulation languages design (except of special constructs for spatial queries). Active rules and triggers can be based on location and need special syntactical constructs to handle spatial data as well.

6 Conclusions

Management of data in the massively distributed environment of *mobile* computing offers new challenging research problems. We have identified those challenges, provided some preliminary solutions and formulated a number of open problems.

Data management issues offers new challenges both at the global, network level as well at the local computing platform of a palmtop computer. The scale of the system and mobility of its parts are unprecedented and the current network infrastructure is simply not capable of providing adequate support.

We have categorized new research problems into mobility, disconnection, design of wireless information services, and energy management. In general, we believe that mobility will have a similar impact on data management as distributed systems had on data management. The fundamental question "centralized or distributed" will now be extended to "static or mobile".

Still the traditional question "if we build it, will they use it?" remains unanswered. The positive answer will critically depend on the applications which will run on the local, palmtop level. Local data management will be the center of activity for the vendors offering new software tools running on the palmtops. To support future applications the lower level systems software will have to be built first; it will handle various levels of disconnection, power management and finally provide new functionalities necessary to access broadcast data.

The case for a wireless connection can be made on the basis of offered flexibility and mobility. What are the "killer" applications that depend on flexibility and mobility? Many suggest "mail enabled applications" such as those targeted by "Notes" from Lotus corporation. These applications will be targeted towards collaboration of users who are on the move. Other classes of applications include local information services (local yellow pages) which will provide a much more detailed service than that of the current telephone yellow pages and application support for the mobile work force. And in general mobile wireless computing could be another added convenience. As in any revolution, mobile computing has its enthusiasts and skeptics. Without taking sides or trying

to make predictions, we conclude that the concept of mobile computing offers challenges and opens new research problems.

7 Acknowledgments

Special thanks go to David Goodman for introducing us to the field of wireless information services and for his continued support and discussions. We would like to thank Arup Acharya, Rakesh Agrawal, Daniel Barbara, Phil Bohannon, Alex Borgida, Dan Duchamp, Marcin Imielinski, Witold Litwin, Ashar Mahboob, and Avi Silberschatz for comments and discussions on the subject of the paper and its presentation. Needless to say any shortcomings or flaws in the paper are the sole responsibility of the authors.

References

[1] Arup Acharya and B. R. Badrinath, "Delivering multicast messages in networks with mobile hosts," To appear in 13th ICDCS, May 1993.

[2] Rafael Alonso and Sumit Ganguly, "Energy efficient query optimization," MITL Technical Report, December 1992.

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

[4] Rafael Alonso, Daniel Barbara, and Hector Garcia-Molina, "Data caching issues in an information retrieval system," ACM TODS, Sept. 1990, pp. 359–384.

[5] Baruch Awerbuch and David Peleg, "Concurrent online tracking of mobile users", Proc. ACM SIGCOMM Symposium on Communication, Architectures and Protocols, October 1991.

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

[7] T. Imielinski and B. R. Badrinath, "Querying Locations in Wireless environments," In Wireless Communications: Future Directions, Kluweir Academic Publishers. October 1992.

[8] B. R. Badrinath and T. Imielinski, "Replication and mobility," In the Proceedings of the Second Workshop on the Management of Replicated Data, November 1992.

[9] B. R. Badrinath, T. Imielinski and A. Virmani, "Locating strategies for Personal Communication Networks," In IEEE GLOBECOM 92 Workshop on networking of personal communications applications, December 1992.

[10] B. R. Badrinath, Arup Acharya, and T. Imielinski, "Structuring distributed algorithms for mobile hosts," In Preparation.

[11] Michael J. Carey, Michael J. Franklin, Miron Livny, and Eugene J. Shekita, "Data Caching tradeoffs in client-server DBMS architectures," Proc. of the 1991 ACM SIGMOD, May 1991, pp. 357–376.

[12] T. Imielinski, Chuan Li, and B. R. Badrinath, "Optimal locating strategies in mobile environments," In preparation.

[13] Danny Cohen, Jonathan B. Postel, and Raphael Rom, "IP addressing and routing in a local wireless network," Manuscript, July 1991.

[14] Daniel Duchamp, Steven K., and Gerald Q. Maguire, "Software technology for wireless mobile computing", IEEE Network Magazine, November 1991, pp. 12–18.

[15] Daniel Duchamp and Neil Reynolds, "Measured performance of a wireless LAN", Columbia University, October 1992.

[16] Neil Fishman and Murray Mazer, "Experience in deploying an active badge system," In IEEE GLOBECOM 92 Workshop on networking of personal communications applications, December 1992.

[17] David Gifford, John Lucassen, and Stephen Berlin, "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.

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

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

[20] Michael J. Franklin, Michael J. Carey, and Miron Livny, "Global memory management in client-server DBMS architectures," Proc. of the 18th International conference, August 1992, pp. 596–609.

[21] John Ioannidis, Dan Duchamp, and Gerald Q Maguire, "IP-based protocols for mobile internetworking," In SIGCOMM 91, September 1991, pp. 235–245.

[22] John Ioannidis and Gerald Q Maguire, "The design and implementation of a mobile internetworking architecture," In USENIX Winter 1993 technical conference January 1993.

[23] James Kistler and M. Satyanarayanan, "Disconnected operation in the CODA file system," ACM Transactions on Computer Systems, Vol 10, No 1, February 1992, pp. 3–25.

[24] "Notes and Vendor independent messaging," Lotus Corporation, Cambridge, MA.

[25] Y. Rekhter and C. Perkins, "Optimal routing for mobile hosts using IP's loose source route option," Internet Draft, October 1992.

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

[27] Carl D. Tait and Dan Duchamp, "Service interface and replica management algorithm for mobile file system clients," Proceedings of the Parallel and Distributed Information Systems Conference, December 1991.

[28] David J. Goodman, "Trends in Cellular and Cordless Communications," IEEE Communications Magazine, June 1991.

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

[30] David J. Goodman and Binay Sugla, "Signalling system draft," Unpublished manuscript.

[31] Kathleen S. Meier-Hellstern, Eduardo Alonso, and Douglas Oniel, "The Use of SS7 and GSM to support high density personal communications," Third Winlab workshop on third generation wireless information networks, April 1992, pp. 49–57.

[32] L J Ng, R. W. Donaldson, and A. D. Malyan, "Distributed architectures and databases for intelligent personal communication networks," Proc of the ICWC, June 1992.

[33] William Lee "Mobile cellular Telecommunication systems," Mcgraw-Hill, 1989.

[34] C. N. Lo, R. S. Wolff, and R. C. Bernhardt, "An estimate of network database transaction volume to support universal personal communication services," Submitted to the 1st International conference on Universal Personal Communications.

[35] 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.

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

[37] M Satyanarayanan, "Scalable, secure, and highly available distributed file access," IEEE Computer, Vol. 23, No. 5, May 1990, pp. 9–21.

[38] D. B. Terry, D. Goldberg, D. A. Nichols, and B. M. Oki, "Continous queries over append-only databases," Proc of the ACM SIGMOD, June 1992, pp. 321–330.

[39] Ouri Wolfson and Sushil Jajodia, "Distributed algorithms for dynamic replicated of data," 11th ACM PODS, June 1992, pp. 149–163.

[40] Hiromi Wada et.al., "Mobile computing environment based on internet packet forwarding," 1992 Winter USENIX, January 1993.

[41] Gio Wiederhold, "Mediators in the Architecture of Future Information Systems", IEEE Computer, March 1992, 25(3), pages 50-62.