A Distributed Directory Scheme for Information Access in Mobile Computers

M.J. Kumar and S. Venkatesh
School of Computing, Curtin University of Technology, GPO Box U1987
Perth 6001 AUSTRALIA
(kumar,svetha)@cs.curtin.edu.au
and
S Panchanathan
Department of Electrical Engineering
University of Ottawa, Ottawa
CANADA
panch@ee.uottowa.edu.ca

Abstract

In this paper, we discuss the design aspects of a dynamic distributed directory scheme(DDS) to facilitate efficient and transparent access to information files in mobile environments. The proposed directory interface enables users of mobile computers to view a distributed file system on a network of computers as a globally shared file system. In order to counter some of the limitations of wireless communications, we propose improvised invalidation schemes that avoid false sharing and ensure uninterrupted usage under disconnected and low bandwidth conditions.

1. Introduction

In the recent past, technological trends in computers, electronics, and VLSI design have led to the decreasing size and increasing power of portable computers [2]. On the other hand, recent trends in telecommunications and widespread prolifiration of computer networks, have led to the use of personal communication systems. More recently, the advent of mobile computers - a marriage of the above two trends has emerged as the focus of interest in research, industry, business and commerce. Mobile computers (MCs) will be used to access commercial data bases such as traffic data, international and domestic news, medical records, banking information, voice mail, and video data bases such as entertainment and educational media. Mobile computing applications demand two types of information availability, i) continuity of coherent access to location-independent files (e.g., bank accounts, email etc.) and ii) access to updated location-dependent information (e.g., traffic data, weather prediction etc.).

Access to information databases in a mobile environment is different from that in stationary computers. A mobile computer's address changes dynamically as its current location affects configuration parameters as well as answers to user queries [3]. In the Internet Protocol (IP) for example, a host IP name is inextricably bound with its network address; moving to a new location means acquiring a new IP address. In distributed computing environments, NFS and AFS ensure access to distributed file systems under normal operating conditions[6]. However, such file systems do not support disconnected operation. Distributed file systems using the client-server model provide security, integrity, transparency, capacity, and shared access of files. The file system in a mobile computing environment should adapt to changing spatial distribution of MCs by dynamic replication of data and services. Transparent access to information files is a very important issue in the development and usage of mobile computers. The coda and odyssey file systems [9] under development by Sathyanarayanan et al. support disconnected operations. In this paper, we discuss the design aspects of a dynamic distributed directory scheme(DDS) that facilitates efficient and transparent access to information files in mobile environments. The proposed interface enables users of mobile computers to view a distributed file system on a network of computers as a globally shared file system.

The paper is organized as follows. In the second section we discuss problems associated with mobile computers and research efforts to alleviate those problems. In Section three we present design aspects of the distributed directory scheme and in Section four we discuss ongoing research

0-8186-7557-8/96 $5.00

and future work.

2. Information Access in Mobile Computers

Mobile computers pose new challenges since they depend on wireless communication as well as wired communication. Wireless communication faces more obstacles than wired communication because the surrounding environment interferes with the signal and introduces noise [4]. Wireless communication is characterized by lower bandwidth, higher error rate, increased latency, and frequent disconnection [1]. In addition, mobility can also cause disconnections. Current wireless networks can cover areas ranging from a few meters to hundreds of Kilo meters with varying bandwidth and support[2]. Owing to their small size and the need to access a wide range of information, mobile computers are inevitably dependent on remote resources[5]. MCs communicate with a base station(BS) or server via a wireless network in order to access files/data. In addition, the BS may be required to copy locally unavailable files/ data from other remote BSs. Some problems associated with mobility are illustrated in Figure 1. When a MC moves from area (or cell) C1 to area C2, the MC is disconnected from the BS in C1 (typically, wireless LAN) and reconnected to a BS in cell 2. As a result the MC has a new host and the new host is required to provide the necessary files/data to the MC. All transfers among BSs are carried out via the wired communication network whereas all communications between MCs and BSs are carried out via the wireless channel. As such the reliability is poor only for communication between MCs and BSs. Information files are likely to be replicated in two or more BSs other than the home repository. The distributed directory enables transfer of requested files from the nearest BS which is currently holding a replica of the requested file/data. Some of the design issues for such a directory in mobile environments are different to those considered in the design of distributed file systems [6], and distributed shared memory systems [7].

Wireless communication faces more obstacles than wired communication. Mobility makes certain data *dynamic* and it may be desirable to have the distributed directories move from one BS to another as MCs shift their geographic locations. In addition, experimental studies show that wireless transmission consumes approximately 10 times more power than reception[3]. Hence, it is desirable to have the BSs transmit directory updates periodically so that MCs can conserve their power. In view of the above constraints, the following factors affect the design of the hierarchical distributed directory.

Directory Size: The directory should be as small as possible since each BS broadcasts changes in its local directory to the MCs in its region.

Diffusion: Portions of directories are moved from one BS to another as MCs disconnect from one BS and reconnect to another. As a result the directory entries are dynamic in nature.

Scalability: Since the number of MC users is bound to increase very rapidly over the forthcoming years, the DDS should support an increase in the number of users with insignificant degradation in performance.

Proximity: The BS should get the requested file from the nearest geographically located BS holding a valid copy of the file rather than the home BS of the file.

The DDS is intended to be scalable in terms of number of users and size of geographical area. In the following section, we present a broad overview of the preliminary version of the proposed DDS.

3. Distributed Directory Architecture

The design of the directory is illustrated in Figure 2. Here, a given geographical area is divided into k subregions or cells 1,2,3,...,k. Each cell has one or more BSs fitted with wireless communication hardware. Now, a MC in cell i ($1 \leq i \leq k$) can connect to one of the BSs in the region i; likewise, if the MC moves to cell j ($j \neq i$, $1 \leq j \leq k$), it will reconnect to a BS in cell j. At the lowest level of the hierarchy, there is a subdirectory at each MC and it is represented by D_{MC}. In other words, the directories at MCs are the leaves of the overall distributed directory. D_{MC} contains information regarding all files currently available on that MC. At the next level of hierarchy is a directory at the BS, D_{BS}. D_{BS} consists of entries (or records) for all files residing in the BS's memory and those available at all MCs connected to that BS. The BSs are interconnected by means of a wired local area network (LAN) or wide area network (WAN). Further, the directories of BSs of the same cell such as C1 are accumulated into one cell (or area) directory such as D_1. Directories D_1, D_2, D_3 . . . , D_k are connected to another BS such as P, which is at a higher level of hierarchy. The directory at P, referred to as D_p contains details regarding all files available in its memory, at its child nodes (D_1, D_2, D_3, . . . D_k), and those available at all the MCs in the regions $1, 2, \ldots, k$. The hierarchical nature of the directory ensures very low access and transfer times. The hierarchical structure can be further expanded to have one or more levels of hierarchy with nodes such as D_p acting as child nodes of another directory at a higher level. Basically, the topology of directories at BSs is that of a k_i-ary tree, where i is the level of the tree. For simplicity we consider k_i=4 for all levels in our examples. To cover large geographical areas, we can build directories with more

levels of hierarchy.

Files are classified as two types: read only files (ROF) and read/write files (RWF). The BS serves as home BS to all RWFs residing in its memory. Each RWF file has a home BS; *writes* and *invalidations* of a RWF file are carried out through its home BS. Each directory at a BS is divided into two sub directories, one for *incoming* and one for *outgoing* files as shown in Figure 3. Information about incoming files includes, the file address, status, and the address of the MC holding a copy of the file. Information about outgoing files includes, the file name, status, and the address of the remote station that requires a copy of the file with RWF status. In both cases, the file address comprises the file name prefixed by the address of the file's home BS. Thus, in the section of the directory related to outgoing files, the home BS of a file contains the addresses of remote BSs that have accessed this file with *read/write* status. Also, in a remote BS, the file address in the incoming part of the directory gives the address of the home BS to which this file belongs. A data file could be in a valid or invalid state. A valid state implies that it is the most recent version of the file whereas an invalid state implies that a more recent version of the file exists elsewhere in the system. All ROF files are valid files. If the status of a RWF is *valid*, the entry for that file in the home BS will link to a *log-entry* for that file. The *log-entry* in turn will point to the address of the file and the update file. On the other hand if the status of a RWF is *invalid*, the entry for that file in its home BS will indicate the address of the *log-entry* for the invalid file. The *log-entry* contains information about the file version, and address of the BS from where the latest version of the file and the update can be obtained. For example, consider that MCi has a copy of file X of version 4, and the latest copy of X is of version 7 and the X version 7 is available in a remote BS. When MCi desires to read (or read/write) the recent most copy of file X, its request probe will include the version number of its copy of X. If the probe encounters a copy of X (either in the home BS or in any ancestor BS) whose version is higher than 4, the updated portion of the file is copied. The ancestor BS will then retransmit the probe with a request to copy versions 5 are greater. Finally, when the probe reaches the remote BS, update portions corresponding to the difference between version 7 and version 5 is copied from the remote BS to the ancestor BS. The ancestor BS upon receipt of the updated portions will append the portions corresponding to the difference of version 5 and version 4 and transmit the cumulative update file to the requesting MC(MCi) via the home BS.

Bifurcation of the directory at each BS reduces the search area for a request depending on whether the request is from a MC or another BS. At regular intervals, each BS broadcasts the directory update information regarding all information files available in its memory and additions/ deletions in directory entries. Each BS compares the version number of its copy of X (even if it is invalid) with that of the request and appropriately changes the version number on the request probe. Likewise, when it receives the update copy from another BS, it will appropriately append update files so that the requesting MC can obtain the updated file. As we go up in the hierarchy, the size of the data base and directory increases. Since all the directories are stored in BSs, the increase in size of directories is not a serious problem. Moreover, caching techniques can be utilized to keep portions of the directory in local caches to enhance performance.

The mobile computing environment poses many research challenges. The dynamic nature of the distributed directory is one of the complex issues to be addressed. When a MC disconnects from one BS (say BS1) and reconnects to another (say BS2), the following directory and file movements are executed:

1) transmit portion of directory (related to the disconnected MC) from BS1 to BS2.
2) transmit all files of disconnected MC from BS1 to BS2,
3) directory changes at BS1 and BS2 should be reflected at their ancestral nodes.

Prior to disconnection, a MC should cache the required files from the BS and write back modified files at a new BS upon reconnection if the file access status is read/write. In the following sections, we discuss *read* and *read/write* operations in detail.

3.1. The *read* Operation

A request originating from a MC or a BS could be a *read* request or a *read/write* request. In the remainder of this section we describe the operations involved with respect to a typical *read* operation. In order to reduce power consumption at the MCs, we use a combination of polling and interrupt requests to establish connection with a BS. Each BS broadcasts a *connectbroadcast* message to all the MCs in its region. The MCs listen to the broadcast and will either connect to the BS or store address of the BS in a *last-in-first-out* stack. At any given time, the top of the stack gives the address of the *most recently available* BS. Once connection is established, the MC transmits a read request to its parent BS. The BS performs the search and supplies the requested file to the MC. As the MC performs only one transmission of request probe the power consumption at the MC is minimal.

In Figure 4, a MC in area A requests for a file X. First, MC will tune into the channel to see if X is available at BS1. If

X is unavailable at BS1, a probe is sent to its parent node D. The probe will propagate through the hierarchy until it encounters an entry for X. Upon receipt of the address of the remote BS containing X, the BS in area A will initiate transfer of X from the remote station to BS1 via the shortest available path on the wired LAN. The file X is transmitted over the wireless LAN from BS1 to MC. The algorithm to perform read operation is shown below.

```
Procedure read(X);
/* At the mobile computers*/
{
  if MC is not connected to a BS then
        if the request is of high priority
           interrupt the most recently available BS
        else wait for a connectbroadcast from a BS
  send request(X) to BS
}
```

Algorithm 1 a

```
/* At the BS on receipt of X*/
{
  if most recent version of X is not available in its memory
        probe (X)
        Transmit (X) to the requesting MC
}
probe (X)
{
  propagate request(X) to the remote BS via the least
                                    common ancestor
          /*remote BS has the most recent version of X*/
  receive(X) from neighboring BS or parent node
}
```

Algorithm 1b

3.2. The *read/write* Operation

Unlike read access, at any given time only one system can obtain read/write access of a file. A RWF file can be obtained with immediate invalidate(*II*) status or delayed invalidate(*DI*) status. If the request is of type *II*, then all copies of requested file should be invalidated immediately, while if the request is of *DI* type, then copies of the requested file are invalidated after the MC issues an invalidate command. When a *read/write* request for file X is made from a MC to a BS, BS will issue a *probe(X)*. The *Probe(X)* will return the address of the closest remote BS that contains X. If X is of ROF type, then remote BS will return an error. If X is of RWF type and the request is *II*, then the remote BS will execute a *invalidate-update* operation. This results in a broadcast of invalidation notice to all remote stations holding a copy of X. The home BS will note the address of the requesting BS in its directory. The remote BS will transmit a copy of the update file to the home BS. If X is a RWF, and the request is *DI*, the file status at all BSs (including BS) holding a copy of X is changed to ROF and the home BS records the address of the requesting BS. In all cases, X or its update file is transmitted from the closest remote station holding a copy of X to the requesting BS via wired LAN and the BS transmits X to the MC via wireless LAN. When the MC completes the write operation of the file, it will issue a *invalidate-update* described later. The BS in turn will transmit X (or update of X) to the home BS and the home BS will issue an invalidation command to all other BSs holding copies of X and change the status of X from ROF to RWF.

4. Performance and Quality of Service

In a distributed environment, it is important to have a scheme that takes minimal time to locate a requested file. In our directory scheme, the file address is interlinked with the address of the home BS. The home BS of any file either has the latest version of the file or the address of the BS holding the latest version of the file. As a result the search time complexity in each directory is O(1). Since logically, the distributed directory has a tree structure of height l, any request probe will visit a maximum of l number of directories. In mobile environments, the battery power should be conserved so that MCs can be used for longer periods. In view of this, we have determined the time taken by the MCs and the BSs to perform various functions such as establishing connections, copying files, initiating invalidations, updating files etc. We have also derived expressions for the energy consumed at MCs while performing the above functions.

Each MC has its own memory, and wireless access to a BSs memory and remote access via the BS to other BSs in the network. Requests for files, may be internal requests: to its local memory, to a file in the BSs memory, or to read/write a file from a remote station. Likewise, a BS may receive requests from MCs in its cell area or from MCs/BSs located elsewhere in the network. MC submits a file request and waits until the response file is received. The MC is in one of two states: busy and waiting.

The time taken for servicing a local file request will depend on the type of processor and memory used by the MC and the size of the file to be transferred. The time for accessing information from host and remote BSs will depend upon such factors as, i) type of connection, ii) type of operation - read or write, iii) size of the file to be copied, v)

bandwidth of wired and wireless networks, and iv) the memory transfer times. For a BS, both local (MCs in its cell area) and external requests arrive at a memory queue. At any given time, only one of them is selected for service on a first-come-first-served (FCFS) basis while the remaining requests are queued at the buffer of the memory. After receiving a request, a BS will either send an acknowledge signal, followed by the requested file to the requesting BS/ MC or to another BS through the wired network depending on whether the request is to a file in it's own memory or to a file in another BS's memory.

The *quality of service* in the context of general multimedia presentations allow control of trade-offs between information loss and presentation timing. We propose that when a read-only access is made to a file, a user should be able to specify a set of parameters that guarantee a certain level of performance that the user is willing to accept and pay for. In the case of text files, there are two relevant issues; first, the version number of the file, and second, the cost incurred in searching for this file and transmitting it over the wireless network. Suppose, a user who has version i of a file in the local memory of the MC makes a request for an update, and the most recent version the file X is n. Now, to obtain the recent most version of X the time and power costs involved may be high. However, the user may not want to wait for this amount of time and/or spend the battery power. On the other hand he/she may settle for an intermediate version (say j, where $i \leq j \leq n$) of the file that entails lower costs. The quality of the file is defined by $\kappa = j/n$. If $\kappa = 1$, the user gets the most recent version of the file and the best quality. If $\kappa < 1$ then the quality of service is poor and it gets poorer as κ reduces; the user gets older versions of files. The quality of service required depends on the type of information file requested. Suppose the user wants to browse a routine news item or a weather report, then a low quality file may be acceptable. On the other hand if the requested information is critical, such as stock rates or traffic patterns then the user may desire a quality of service of $\kappa = 1$.

5. Discussion

The distributed directory scheme will be evaluated to determine its 1) reliability, 2) scalability, 3) portability, 4) ability to maintain coherence, and 4) speed. There are three critical time parameters involved with information access in MCs[5]; 1) Access time: time elapsed between the initiation of a client query and the receipt of the response, 2) Transfer time: time elapsed to transfer an information file from a remote BS to the MC, and 3) Tuning time: time spent by the MC listening to the channel. MCs are subjected to the limitations of wireless communications: low bandwidth, disconnected operation, limited battery power, and high error rates. The directory architecture proposed in this paper enables continued operation under disconnected conditions and minimizes transmission from MCs in order to avoid high power consumption.

To validate our theoretical findings, we are carrying out simulation studies under various network conditions. In most file sharing instances with regard to RWF files, a client with read/write access modifies only a small portion of the accessed file. As a result it is sufficient if the file update mechanism, modifies the affected portion of the file rather than the entire file. This would result in enormous reduction in communication overheads and hence power consumption. Further, invalidation of a file actually means locking the file until next update. The file is not purged at any BS, it is updated when the network is not busy or when there is a request for the file at the BS. This improvised *invalidate-update* procedure reduces overheads significantly. On the other hand purging of files at MCs on a priority basis has a negative effect on *invalidate-update* procedure. Our ongoing simulation studies and future research is directed to address these issues.

References:

1. E. Buitenwerf, G. Colombo, H. Mitts, and P. Wright, UMTS: Fixed Network Issues and Design Options, IEEE Personal Communications, February 1995, pgs. 30-37.

2. D. Dunchamp, S.K. Fiener, and G.Q. Maguire, Jr., Software Technology for Mobile Computing, IEEE Network Magazine, November, 1991, pgs. 12- 18.

3. G.H. Forman and J. Zahorjan, The Challenges of Mobile Computing, IEEE Computer, April 1994, Vol. 27, No.4, pgs. 38-48.

4.D. J. Goodman,Trends in Cellular and Cordless Communications, IEEE Communications Magazine, June 1991, pgs. 31-40.

5. T. Imielinski and B.R. Badrinath, Wireless Computing, Communications of the ACM, Vol. 37, No. 10, October 1994, pgs. 19-28.

6. J.J. Kistler and M. Satyanarayanan, Disconnected Operation in the Coda File System, ACM
Transactions on Computer Systems, Vol. 10, No. 1, February 1992, pgs. 3-25.

7. M.J. Kumar, S. Venkatesh, D. Kieronska, and L.M. Patnaik, Hierarchical Directory-Based Shared Memory Architecture, The Computer Journal, Vol. 38, No. 3, 1995, pp. 207-216.

8. L.B. Mummert M.R. Ebling, and M. Satyanarayanan, Exploiting Weak Connectivity for Mobile File Access, In Proc. of the Fifth ACM Symposium on Operating Systems Principles, Copper Mountain Resort, Colorado, Dec. 1995.

9. M. Sathyanarayanan, Mobile Information Access, IEEE personal Communications, Vol.3, No. 1, Feb. 1996.

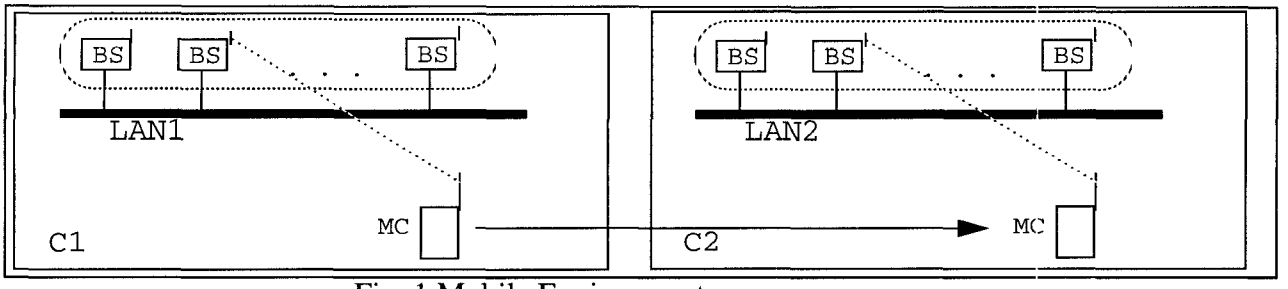

Fig. 1 Mobile Environment

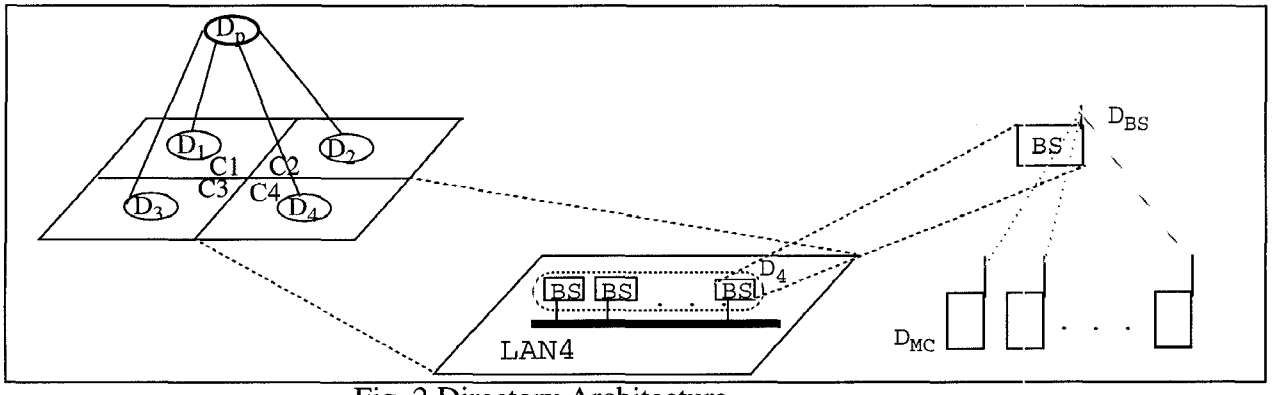

Fig. 2 Directory Architecture

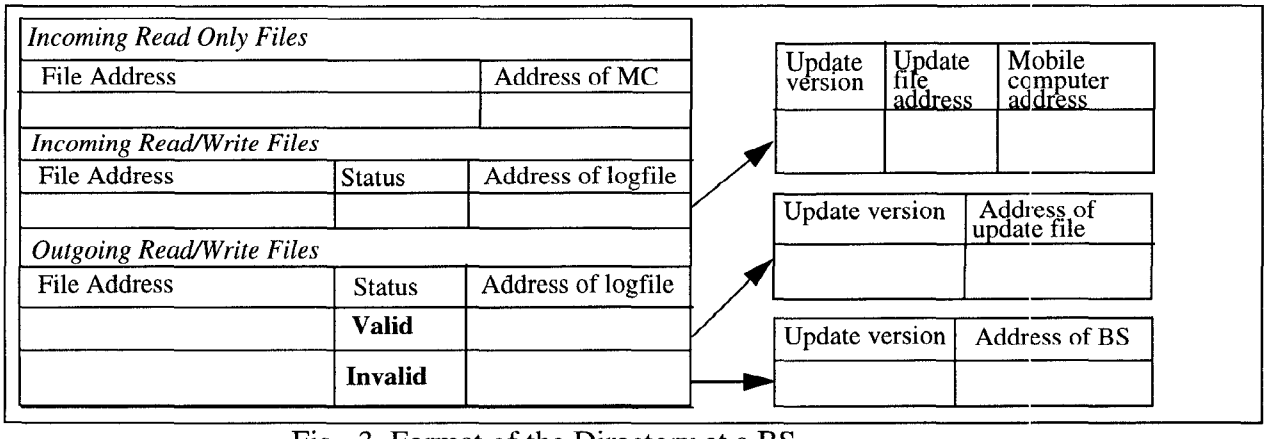

Fig . 3 Format of the Directory at a BS

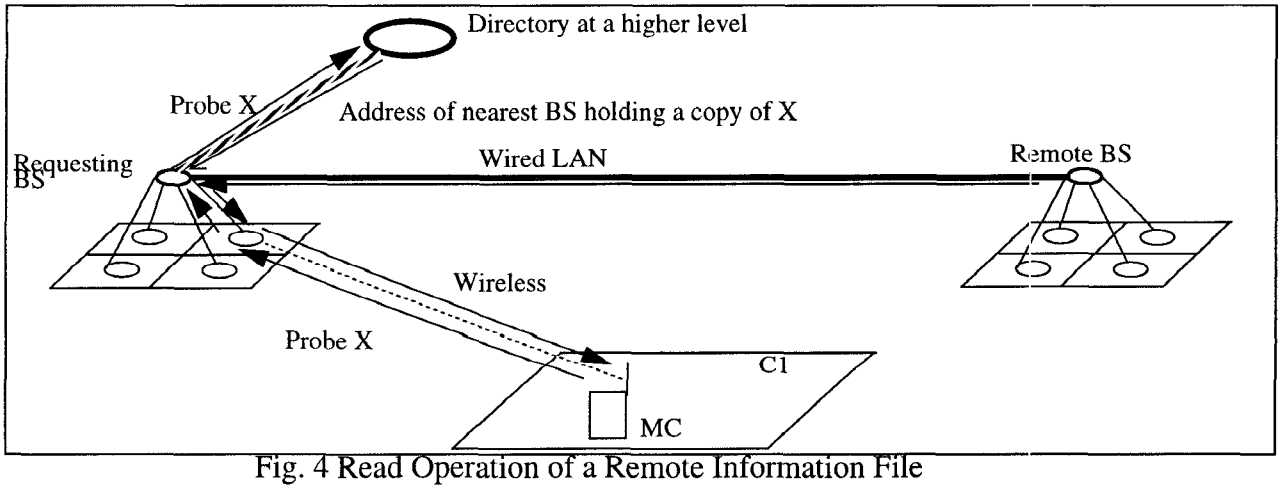

Fig. 4 Read Operation of a Remote Information File