Design and Implementation of a Peer-to-Peer Data Dissemination and Prefetching Tool for Mobile Users

Maria Papadopouli Henning Schulzrinne
Department of Computer Science
Columbia University
New York, NY 10027
e-mail: {maria,hgs}@cs.columbia.edu

Abstract

This paper presents 7DS, a novel peer-to-peer data sharing system. Peers can be either mobile or stationary (such as info-stations). 7DS is an architecture, a set of protocols and an implementation enabling the exchange of data among peers that are not necessarily connected to the Internet. Communication is typically, but not necessarily, wireless. 7DS runs as an application and communicates with other 7DS participants via a LAN. It operates in two modes, namely prefetch and on-demand. In the prefetch mode, it anticipates information needs of users and the system queries other peers for these data. In the on-demand mode, the user can directly search for information among peers. 7DS can work complementary to other data access methods. 7DS enhances the collaboration by allowing users to create on-the-fly an ad-hoc network and browse the content of the cache of peers that have been made accessible to it. It can be used to spread information and for data access, particularly, for location-dependent data (such as weather or traffic reports, news, tourist guides, campus events or news) and popular data (music files, news, video games) that do not change very rapidly. We discuss the design and an implementation of 7DS.

1 Introduction

New services offered by telecommunication companies and content providers which expose consumers to spatial information have already arrived. Such services provide news, traffic or weather reports, maps, guide books, music and video files, games, notifications about changes in environmental conditions [1] and points of interest. The information is stored in repositories, such as web servers, database servers or more specialized repositories such as query-by-image-content databases or geographical information systems. Access to information will become as important as voice communications for wireless roaming through metropolitan areas [2, 3, 4].

We separate mobile information access methods in to three main categories. The first approach provides "continuous", wireless Internet access, such as CDPD, 3G wireless, 802.11 and two-way pagers. The second approach provides information access via fixed (stationary) information servers or *info-stations* in local geographic proximity [5, 6, 7]. The *info-stations* are "information kiosks", located at traffic lights, building entrances and airport lounges. These two approaches are using an infrastructure. If the wired infrastructure is low-bandwidth, they can be combined by having caches at the base stations. Throughout the paper we use the term *info-station* for any information server or base-station (with or without cache), that provides Internet or data access to mobile users. The third approach is without the support of any infrastructure (i.e., ad hoc) based on peer-to-peer data sharing among the mobile, wireless devices.

Current wireless Internet access either have sparse coverage and low-cost and high speed (802.11) or have major-cities-only coverage, high cost (Metricom [8]) or have wider coverage, but extremely low rates and high costs (CDPD, RIM). Also, info-stations offer high speed but discontinuous coverage. Given the exceedingly expensive license fees attained in recent government auctions of spectrum, the bandwidth expansion route is bound to be expensive. Similarly, the cost of tessellating a coverage area with a sufficient number of base stations or info-stations coupled to the associated high speed wired infrastructure cost is forbidding. For the next few years, continuous connectivity to the Internet will not be universally available at low cost.

We propose $7DS$, a new peer-to-peer solution that complements these existing approaches. Peers can be either mobile or stationary (such as info-stations). $7DS$[1] is an architecture, set of protocols and implementation enabling the exchange of data among peers that are not necessarily connected to the Internet. $7DS$ runs as an application and operates in two modes, namely prefetch and on-demand. In the prefetch mode, it anticipates information needs of users. In the on-

[1] "$7DS$" stands for "Seven Degrees of Separation", a variation on the "Six Degrees of Separation" hypothesis, which states that any human knows any other by six acquaintances or relatives. There is an analogy with our system, particularly, with respect to data recipients and the device with the "original" copy. We have not explored if a similar hypothesis is true here.

demand mode, it searches for information among peers when the user searches for information or requests a URL.

$7DS$ can also run on the top of location-dependent services and prefetch data on behalf of them, when the host cannot access data via *info-station*. $7DS$ is particular useful for popular information that do not change rapidly (and therefore do not require continuous Internet connectivity). Examples of such information are news, traffic or weather reports, maps, guide books, music and video files, games. In the context of these services, spatial locality implies locality in the data that mobile users want to access and in their access patterns. We anticipate that there will be a high probability that the data a mobile host queries, can be found in the cache of another mobile host in close geographic proximity. The system exploits the high spatial locality of information in pervasive computing environments and also the fact that mobile users are likely to be more flexible in their information tastes, (media) quality and information accuracy requirements.

In such a network of mobile hosts, each new device contributes to an ever denser web of communication, where data can move from subway rider to subway rider, among anonymous persons meeting each other in the streets, in the hallways of an office building, at a conference, a public area (such as a train or airport platform), in a battlefield situation or in a disaster recovery area with rescue teams. Today, it is cumbersome to set up an ad hoc network that enables people to share data with each other. $7DS$ allows users to browse the content of the cache of peer that has been made accessible to it. Participants in $7DS$ obtain URLs, web pages, or any application specific data of modest size, cache them and exchange them with others who are interested in them. Each device maintains a cache containing information items received by pervasive computing devices.

This system raises several interesting research questions; some are theoretical, such as study of the data propagation using epidemic models, percolation theory and game theory, and others design-related. In [9] we evaluate via extensive simulations the effectiveness of our system in data dissemination and prefetching for mobile devices. Also, we investigate the effect of the wireless coverage range, network size, query mechanism, cooperation strategy among the mobile hosts and power conservation with a very large number user mobility scenarios. The contribution of this paper is the design and implementation of $7DS$. The remainder of this paper is organized as follows. Section 2 gives an overview of the main components of $7DS$, Section 2.2 discusses the design decisions and implementation. In Section 3, we discuss related work. Finally, we present conclusions and describe future research directions in Section 4.

2 System architecture

2.1 Overview

In Section 1, we introduced $7DS$; an architecture, a set of protocols and an implementation enabling peer-to-peer data sharing. The mobile host has a network connection to access the Internet, e.g., via a wireless modem, a base station, or a Bluetooth device and is also capable of communicating with other hosts via a wireless LAN. $7DS$ runs as an application on mobile hosts and communicates with other $7DS$ participants via a wireless LAN. When the mobile host experiences intermittent connectivity to the Internet or there is no *info-station*, $7DS$ queries other peers in close proximity via the wireless LAN to acquire the data. Figure 1 illustrates how $7DS$ operates.

The system consists of four main components: the data access engine (DA), a cache manager (CM), a search engine (SE) and a decision maker (DM) attached to several monitors (Figure 2). The main operations of the data access engine are querying and the complete data exchange. Applications run on mobile hosts requesting data from application-specific servers. When there is a loss of the Internet connection, these requests are queued up locally. We call this list of requests generated by the applications (such as web, tour guide, traffic reports) a *prefetching plan*. An example of a prefetching tool reads the history file ("netscape.hst") of the user [2] and tries to predict the URLs the user will visit in the next few hours. It, then, appends these URLs in the prefetching plan. $7DS$ forms a query which includes all these queued requests and multicasts it periodically via the wireless LAN. The $7DS$ peers listen to a well-known multicast group. The Query Scheduler Bcast (Figure 2) removes duplicate or expired requests. At the same time, $7DS$ QueryReceiver listens for queries on the multicast address. As we describe in more detail in the following section, each request has an identifier. For each request, it searches the cache. In the case of a cache hit, it forms and broadcasts a report. The report is a short description of the relevant data in the local cache of a mobile host that has received the query and is willing to share its data. The $7DS$ report receiver collects these reports from all hosts that responded and selects the most relevant based on some application-specific criteria. It waits for an interval of time before selecting the most relevant reports for which to get the complete cached data (that correspond to the selected report).

2.2 Design and implementation

Based on our target environment we identify the following design goals for $7DS$.

- Reusability of existing components and data neutrality: Instead of inventing a new technology or new protocol,

[2]Each Netscape Navigator user has his/her own history file called "netscape.hst".

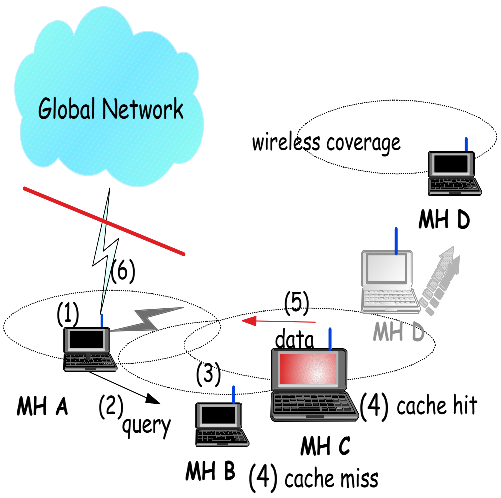


Figure 1: An example of the communication among $7DS$ peers.

$7DS$ uses IP multicast for querying the peers, HTTP for receiving the complete data and existing software for searching and indexing a cache. An important feature of our architecture is its easy deployment. The system displays both the reports as well as the complete data information using any browser.

- Transparency to wired and wireless networks: Except for increased object retrieval latency, the user experience when connected to the Internet and when in ad-hoc mode is the same.
- Easy configuration: $7DS$ should configure itself with minimal manual intervention and the system does not require any registration for data distribution.
- Robustness: the system must be resilient to failures and inconsistencies that occur in this very dynamic environment.
- Expressiveness: $7DS$ should be flexible to support different applications and be able to form queries for their prefetching plans as well as incorporate application-specific criteria for the selection of the appropriate cached copies.
- Resource awareness and adaptivity: $7DS$ trades power for data only when the battery level is above a threshold. Via a battery monitor and a power management protocol, $7DS$ aims to adapt communication to reduce energy consumption during idle periods, when there is low expectation for data or collaboration and when the battery life is below a threshold.

We chose XML to describe the client queries and the reports. XML allows the encoding of arbitrary structures of hierarchical named values; this flexibility allows information providers to add application-specific information. $7DS$ extracts the metadata from the queries received from other peers and performs an attribute-matching search in its local cache. A report includes the query body and the relevant information in the local cache of the peer that responds.

In the current implementation (shown in Figure 2) a data object is identified by a URL. The report includes the local and original URL of the object, time the object was cached locally, and time the original object was created, as well some optional information, such as, its title, size, format and summary. The local URL indicates the object in the cache of the peer that sent the report. For example, $7DS$ peer 10.4.1.5 has a copy of the page *"http://www.boston.com/index.html"* in its local directory *"www.boston.com"*. The local URL of that copy is *"http://10.4.1.5:9000/www.boston.com/index.html"*. A request is identified by two fields, the URL and the time it was created. For each received query, $7DS$ extracts the requests. It ignores the duplicate or "old" requests and for the rest, it gets the URLs and the keywords (if they are specified) and searches the index. $7DS$ logs all the recently served requests (i.e., requests for which it searched the cache). A request in the log is considered "old". In the case of a cache hit, if the cached object has the right permission, report sender creates a report and multicasts it. Currently, we use *glimpseindex* for indexing the cache and *Glimpse* [10] to query the cache.

The report selector periodically monitors the received responses and chooses a report based on application specific criteria. It extracts some metadata, such as the URL to the cached copy of the peer and its summary, and passes it to the $7DS$ web client for display. The web client issues an HTTP get request using the local URL of the selected report to receive the complete object. A miniature web server is running as a part of $7DS$ which services the HTTP get requests. The primary information propagation is through the use of caching rather than reliable state maintenance. It is not a goal of the current prototype to resolve inconsistency among copies of a data object. 7DS peers may have several objects matching the query criteria.

To optimize bandwidth and power consumption, we allow queriers to filter the reports returned and only ask for a subset of the objects referenced. The filter application can take into account the original URL, the author, language, size and content type of the object, as well as the geographic location the object is associated with. Some of this information is inherently provided by web objects, while others require adding additional meta information. By default, the most recent object is selected.

CM organizes and indexes the cache. Through a GUI, it provides the capability to the user to view, browse and manage the cache. In the current prototype, the content of the cache is displayed in a tree-like structure. We are extending it to

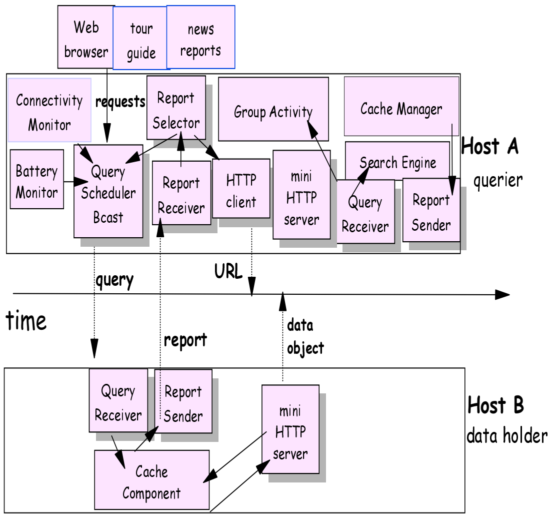


Figure 2: Main components of the architecture in the current $7DS$ prototype. The complete $7DS$ appears on the host A whereas for B only the components that interact with A are shown. The data access engine consists of the components with the shadow.

support grouping of the cache content to predefined categories and searching tools using the meta-data attributes of the stored objects. As part of managing the cache, the user can set the access permissions for files and directories in the cache and specify the objects to share with other peers. The user can also explicitly add or remove files. Periodically, CM removes the expired objects, updates the index with these changes and also includes newly cached objects. Apart from deleting the expired objects, the system may try to prefetch them again transparently. Through the GUI, the user marks which pages need to be prefetched when they expire. When CM searches the cache to remove the expired objects, for each marked page, it forms a request and adds it in the prefetching plan.

In particular, $7DS$ supports the following operations for querying, cooperation and power conservation:

***Active* and *passive* querying**

In active mode, $7DS$ periodically checks if there are requests queued up locally and broadcasts them. In passive mode, an advertisement "triggers" the querying. Essentially, the advertisement notifies $7DS$ peers for the presence of a $7DS$-enabled *info-station*. Devices with limited power capabilities use the "passive" mode to participate only when the expectation for data availability is high. Like the reports, the advertisements are also written in XML and include one or multiple fields. A typical advertisement has one field. However, it could contain several fields with description of the *info-station* content. Upon its receipt, $7DS$ in an attribute matching fashion filters it or displays it for browsing.

Data sharing* and *forwarding

Data sharing and forwarding are the two facets of cooperation among $7DS$ peers. Data sharing takes place in response to queries of other peers. In the case of local cache hits, $7DS$ broadcasts a report with the relevant information. The motivation for the forwarding is to "extend" the wireless coverage of the $7DS$ communication. Depending on the forwarding option, a peer rebroadcasts the query, report or advertisement. This rebroadcasting or flooding often results in an implosion due to the "blind" broadcast of a message, regardless of whether or not a node may have already received the message by another neighbor. To avoid these problems and still keep a simple, stateless approach, we adopt a simple method in which, in addition to TTL in the forwarding messages, $7DS$ restricts the rate it forwards messages and responds to queries.

Power conservation

When enabled, the mobile host periodically turns off its wireless LAN interface. $7DS$ periodically checks the battery level via the *apm* daemon and adapts its collaboration accordingly. In particular, the system adapts its communication with other $7DS$ peers by tuning several "thresholds" in the battery level. For example, it may set three values of the battery: when the battery level is above the highest value, the system can fully collaborate. Within the highest and second highest value, the system only applies a partial participation in the system. Below the third value, the system stops participating in the $7DS$ network. Generally, the degree of participation depends on parameters such as type of querying (active or passive, frequency interval) and type of collaboration (data sharing and forwarding support). $7DS$ is engaged entirely in the participation when it both actively and passively queries, and supports data sharing and forwarding. In a partial participation, it adjusts the forwarding and querying intervals and depending on its energy constraints, it may switch from active to passive querying. The default setting is as follows: for battery level above 75%, $7DS$ uses data sharing and active querying, when it is within 50% to 75%, it switches to passive querying, and when the battery level falls bellow 50%, it stops participating in the system. Via a GUI, the user can change this setting.

$7DS$ records the identifiers of the queries it served in the last time interval. $7DS$ logs timestamps of the reports and responses to HTTP get requests it sends. A service provider may encourage cooperation via $7DS$ by providing financial incentives to "good" $7DS$ users. $7DS$ may upload this log to a central server as indicator of their sharing activity.

The prototype is written in Java (not including the Glimpse code that is written in C.). In Appendix A, we include some snapshots of the main GUI (running $7DS$), part of the cache manager (cache viewer) and the configurator. We have tested the actual system (using web browser) as a prefetching tool in Linux. Currently, we intergrate $7DS$ with a tour guide and an academic news notification system to deploy it in the campus. In addition, we are in the process of implementing it on

Windows CE.

3 Related work

Napster [11] and Gnutella [12] are two systems that explore the cooperation among hosts and enable data sharing among users in a fixed wired network. The first is focus in sharing music files, whereas the latter for any type of files. In the case of Gnutella, the hosts need to maintain fix connections with each other. This would not be easy or feasible in the very dynamic setting we are considering. Unlike Gnutella, $7DS$ does not need to discover its neighbors or maintain connections with them, but only multicast its queries to a well known group. Unlike Napster that requires a centralized server for indexing the music files to be shared, $7DS$ operates in a distributed fashion without the need of any central server. Moreover, Napster requires user intervention and effort for uploading files, whereas $7DS$ does this automatically.

In earlier work [13], we investigated a different facet of cooperation, namely network connection sharing. Mobile devices with multiple wireless interfaces can serve as temporary gateways to wide-area wireless networks.

Info-stations have first been mentioned by Imielinski in the DataMan project [5]. Badrinath was among the first to propose infrastructure for supplying information services, such as e-mail, fax and web access by placing info-stations at traffic lights and airport entrances.

Caching and prefetching have been successfully employed to alleviate user perceived latencies and there has been extensive research. In the context of mobile users, hoarding is a similar technique to prefetching to improve the data availability (for users that experience intermittent connectivity) [14, 15]. Prefetching targeted for mobile users in a wide area wireless network has been used in [16] in a context more similar to ours. Tao Ye *et al* assume an info-station deployment. They consider data representation in different levels of detail. Their prefetching algorithm uses location, route and speed information to predict future data access. Their emphasis is on devising and evaluating techniques for building network-aware applications. They describe an intelligent prefetching algorithm for a map-on-the-move application that delivers maps, at the appropriate level of detail, on demand for instantaneous route planning. When a mobile user enters an info-station coverage it prefetches a fixed amount of k bytes (that corresponds to a map with a certain level of detail). The amount k depends on its user speed. They investigate the effectiveness of info-stations as compared to a traditional wide-area wireless network.

Kravets *et al* [17] present an innovative transport level protocol that achieves power savings by selectively choosing short periods of time to suspend communication and shut down the communication device. It queues data for future delivery during periods of communication suspension, and decides predicting when to restart communication. This work motivated us to consider schemes for predicting high data availability in our setting to power on the communication device and start $7DS$.

Relevant to ad hoc networks is the on-going research on sensor networks. Heinzelman *et al* [18] present a protocol for information dissemination in sensor networks. In their setting, the sensors are fixed and the network fully connected. They measure both the amount of data these protocols disseminate over time and the amount of energy the dissipate. It features meta-data negotiation prior to data exchange to ensure that the latter is necessary and desired, eliminating duplicate data transmissions, and with power resource awareness. They compare their work with more conventional gossiping and flooding approaches.

$7DS$ is an information discovery mechanism. From this sense, it is similar to some resource discovery systems such as Sun's Jini, SLP [19] and INS [20]. Our system has some features in common with these works (e.g., like INS, it uses XML-based description of the resources), but differs in the overall architecture and target environment (e.g., usage of the system, type of queries, and hosts). These protocols primary focus is the routing of resource discovery requests. Instead, this work addresses different research issues, e.g., the effectiveness of the protocol with respect to power conservation, different mobility patterns and collaboration strategies.

4 Conclusions

In this paper, we presented $7DS$, a new peer-to-peer data sharing system. $7DS$ is an architecture, a set of protocols and an implementation enabling the exchange of data among peers that are not necessarily connected to the Internet. It runs as an application complementary to other data access approaches (such as via base stations or info-stations). It anticipates the information needs of users and fulfills them by searching for information among peers.

We presented the design and implementation of $7DS$. Our current research direction includes the intergration of $7DS$ with a tour guide and an academic news notification system and its deployment in the campus. It is part of future work to investigate prediction algorithms for data availability and access and use them to improve the power utilization.

A $7DS$ Snapshots

References

[1] ALERT. http://www.alertsystems.org.

[2] Airflash. http://www.airflash.com.

[3] Cell-Loc. http://www.cell-loc.com.

[4] "NTT DoCoMo." http://www.nttdocomo-inpaku.com.

[5] DATAMAN. http://www.cs.rutgers.edu/dataman/.

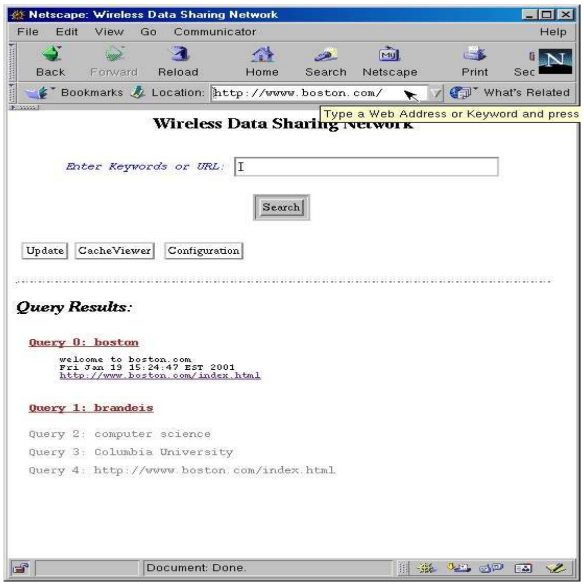
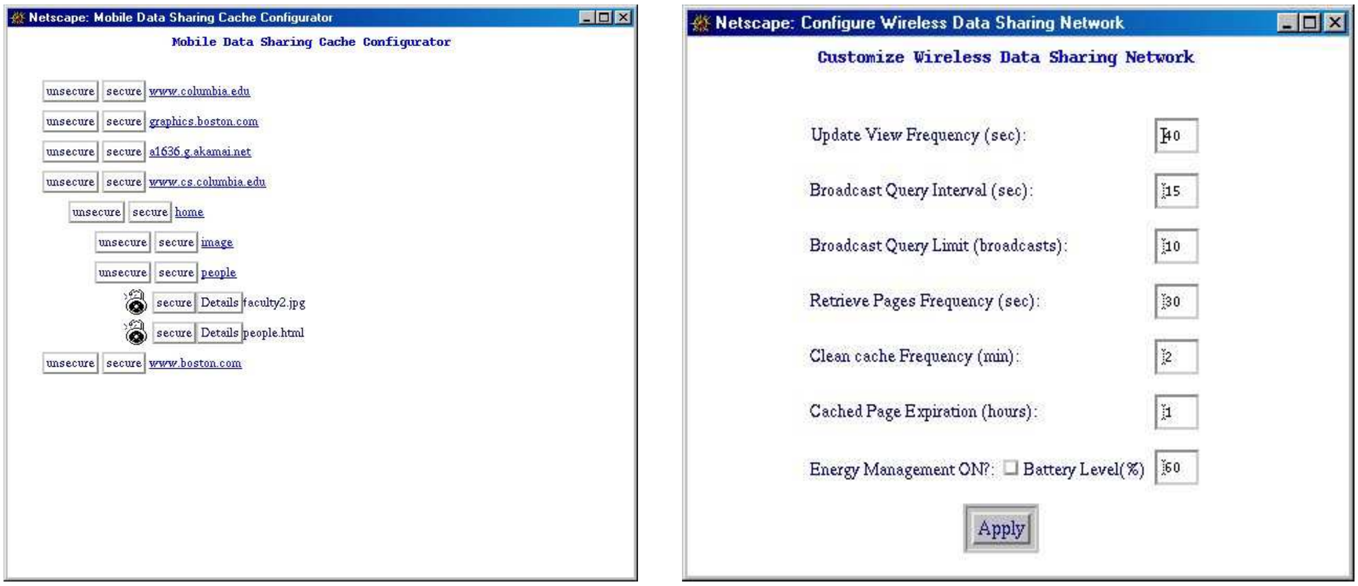


Figure 3: $7DS$ main GUI. In the upper part of the GUI the user can enter a URL or form a keyword based query or view the cache manager or configura tor. In the lower part, there are the query results. Some queries are still pending (e.g., 2, 3 and 4). There are responses for query 0 ("expanded", i.e., showing the report) and for query 1.



(a.1) (a.2)

Figure 4: (a.1) is a part of the cache manager GUI for setting up the permission of the cached objects for sharing with other peers and (a.2) is the configurator GUI.

[6] R. H. Frenkiel, B. R. Badrinath, J. Borrs, and R. D. Yates, “The infostations challenge: balancing cost and ubiquity in delivering wireless data,” *IEEE Personal Communications Magazine*, vol. 7, pp. 66–71, Apr. 2000.

[7] N. Davies, K. Mitchell, K. Cheverst, and G. Blair, “Developing a context sensitive tourist guide,” 1998.

[8] Metricom, “The Ricochet wireless network overview,” 1999.

[9] M. Papadopouli and H. Schulzrinne, “$7ds$: A peer-to-peer data dissemination and prefetching tool for mobile users,” in *submission for publication*.

[10] Glimpse. http://www.webglimpse.org.

[11] Napster. http://www.napster.com.

[12] Gnutella. http://gnutella.wego.com.

[13] M. Papadopouli and H. Schulzrinne, “Connection sharing in an ad hoc wireless network among collaborating hosts,” in *Proc. International Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV)*, (Basking Ridge, New Jersey), pp. 169–185, June 1999.

[14] J. J. Kistler and M. Satyanarayanan, “Disconnected operation in the coda file system,” *Thirteenth ACM Symposium on Operating Systems Principles*, vol. 25, no. 5, pp. 213–225, 1991.

[15] H. Lei and D. Duchamp, “An analytical approach to file prefetching,” in *USENIX Annual Technical Conference*, (Anaheim, CA), Jan. 1997.

[16] T. Ye, H.-A. Jacobsen, and R. Katz, “Mobile awareness in a wide area wireless network of info-stations,” in *Proc. of Mobicom*, (Dallas, Texas), Oct. 1998.

[17] R. Kravets and P. Krishnan, “Power management techniques for mobile communication,” in *Proc. of Mobicom*, (Dallas, Texas), pp. 157–168, Oct. 1998.

[18] J. Kulik, W. Rabiner, and H. Balakrishnan, “Adaptive protocols for information dissemination in wireless sensor networks,” in *Mobicom* (ACM, ed.), (Seattle, Washington), pp. 24–35, Aug. 1999.

[19] C. Perkins and E. Guttman, “DHCP options for service location protocol,” Internet Draft, Internet Engineering Task Force, Feb. 1999. Work in progress.

[20] W. Adjie-Winoto, E. Schwartz, H. Balakrishnan, and J. Lilley, “The design and implementation of an intentional naming system,” in *Proc. ACM Symposium on Operating Systems Principles*, (Charleston, South Carolina), pp. 186–201, Association for Computing Machinery, Dec. 1999.