Seven Degrees of Separation in Mobile Ad Hoc Networks *

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

Abstract

We present an architecture that enables the sharing of information among mobile, wireless, collaborating hosts that experience intermittent connectivity to the Internet. Participants in the system obtain data objects from Internet-connected servers, cache them and exchange them with others who are interested in them. The system exploits the fact that there is a high locality of information access within a geographic area. It aims to increase the data availability to participants with lost connectivity to the Internet. We discuss the main components of the system and possible applications. Finally, we present simulation results that show that the ad hoc networks can be very effective in distributing popular information.

1 Introduction

In a few years, a large percentage of the population in metropolitan areas will be equipped with PDAs, laptops or cell phones with built-in web browsers. Thus, access to information and entertainment will become as important as voice communications for wireless users. However, current wide area network wireless deployment is characterized by intermittent connectivity, low bit rates, high cost, and high end-to-end delays. For the next few years, continuous connectivity to the Internet will not be universally available, at least at low cost. In some environments, such as subways or planes, wireless connectivity will remain poor or non-existent for the foreseeable future. We anticipate that mobile computing devices will be equipped with low-power, low-range (and in some cases high-speed) wireless connectivity such as BlueTooth [1] or IEEE 802.11 [2]. While primarily conceived as access technologies for landline networks, these wireless media and the devices equipped with them can be used to build networks that require little or no fixed infrastructure. In such a network, each new device contributes to an ever denser web of communication, where data can move from subway rider to subway rider, between anonymous persons meeting each other in the streets, in the hallways of an office building, a conference between colleagues, a public area (e.g., a train or airport platform), in a battlefield situation or in a disaster recovery area with rescue teams.

*This work was supported by an IBM UPP grant and a New York CAT grant as well as a DURIP equipment grant and equipment donated by Microsoft.

Such mobile devices can obtain data from other mobile devices, but also from fixed servers, sensors or gateways to the Internet. For example, in a university campus setting, as a student is walking, she may receive the schedule of the talks in a conference room, or the lectures in a classroom. When she approaches the cafeteria, she receives today's menu, and later as she passes through the cultural center, the event schedule. During her walk she may meet other students and exchange information.

We describe and evaluate a system, 7DS[1], that facilitates such information exchange for mutual benefit. Participants in 7DS obtain data objects from Internet-connected servers, cache them and exchange them with others who are interested in them. The information exchanged can take the form of web pages, maps, short movie clips, sound clips or any other data object of modest size.

Sharing data objects is one example of how wireless devices can cooperate for mutual benefit. In earlier work [3], we had investigated another aspect, namely, how owners of mobile devices with multiple wireless interfaces can serve as temporary gateways to wide-area wireless networks. Here, we explore a different facet of cooperation, namely the data sharing, in which devices in close geographic proximity can communicate with each other.

Sharing data comes at a cost for those possessing the data, as they need to expend battery power to transmit it to strangers. If battery power is precious, we cannot just rely on the altruism of random strangers to make the system work. This problem can be addressed by embedding a barter system into 7DS, as discussed further in Section 2.

7DS exploits the fact that there is a high locality of information access within a geographic area. For example, it is likely that many subway riders will want to read the day's paper or watch the current headline news or get the weather report. Also, people on the move are likely to be more flexible in their information tastes. If the alternative is looking at subway ads for divorce lawyers, a *New York Times* reader may well appreciate the *New York Post* sport's pages. This system also lends itself to distributing information to tourists, for example, maps and information about local sights.

Each device maintains a cache containing information items received either from the landline network, stand-alone servers such as information kiosks, any wide-area wireless network or other *7DS* participants. Cache items are marked up with context attributes such as display requirements, media format, general topic area such as "news" or "weather", and access restrictions. Queries consist of a combination of attributes and text searches.

Our work has a superficial similarity to ad hoc wireless networks. However, while ad hoc networks aim to build a continuous packet forwarding mechanism between participants and thus focus on routing protocols, 7DS is a store-and-forward system for application data units, not packets, with no assumption of real-time connectivity or a continuous flow of data. As we describe in Section 2.1, instead of routing or packet forwarding, the system "manages" forwarding via a pull and push-based data transmission.

In this paper, we focus on modeling the data propagation in such an environment. We investigate

[1]The expression "six degrees of separation" is the hypothesis that any human knows any other by six acquaintances or relatives.

two representative user mobility patterns that reflect how mobile users interact in a dense urban area: one models a subway, the other pedestrians “randomly” walking through the city. The subway model differs from the random-walk (or randway) model [4] in that there is movement of groups of users (e.g., from a platform to the train), also interactions between a group of users (e.g., inside the car or in a platform) and make simple assumptions about the wireless transmission based on coverage and layout (e.g., no transmission get through a car). As a third model of information propagation, we also analyze and simulate an environment where information propagates according to a simple epidemic model, as it is analytically tractable.

In all cases we are interested in the number of data holders at the end of a train trip or walk, the average time that a mobile host has to wait until it receives the data and the average delay for the message to spread among the users as a function of the popularity of the data and the frequency which which *7DS* participants query for data.

In our evaluation, we focus on a single data object. This may be either a single web page or a set of objects with common attributes, so that the participants see them as equivalent. In all of our models, we assume that initially, just one mobile carries this data object. It is likely that many more wireless devices will be “seeded” with objects of interest, so that our performance number represent an upper bound on the data spreading delay.

Participants that do not have the data object of interest query for it periodically, with the query period as a major parameter of our investigation. The popularity of a data item is represented in our models by the number of queriers in a particular geographic area.

In the subway scenario, we specify a time interval. The participants of the system are the users interested in a data item that arrived at a subway stop during that interval. We measure the data propagation on the platform or in the train among the participants during that interval. The population is the number of the participants in the system. By the end of the trip, for a population size of 50 users 78% obtains the data. This population size corresponds to an interarrival time of 60 s (of one new person) at each stop. For this, we consider only the users that are interested in the data item. The interarrival time is used as an indicator of the popularity of that data. Whereas only 63% receives the data by the end of their trip, for a population size of 17 users [2]. Our simulations show that the ad hoc networks can be very effective in distributing popular information. In Section 2.2 we describe the results in more detail.

The remainder of this paper is organized as follows. Section 2.1 gives an overview of the infrastructure and the main components of *7DS*. Section 2.2 presents simulation results. Then, in Section 2.3, we describe the simple epidemic model of the data propagation with the analytical and simulation results.

[2] In this case the interarrival time at each stop is 180 s.

2 Data Sharing

2.1 Overview of the Architecture

In *7DS* mobiles access sporadically the Internet via a wireless WAN. A mobile host may also communicate with either other mobiles or proxy servers in close proximity via a wireless LAN. The proxy servers have larger storage, power supply and processing capabilities than the mobiles. We assume a well known multicast address in which the cooperating mobiles or the proxy servers may listen and send requests and responses. The requests are queries for a network connection or for some information. As we mentioned in Section 1, a host that has lost connectivity to the Internet may request from another host to act temporarily as gateway for a given flow. In [3] we describe in detail the architecture for the connection sharing.

Data sharing can be pull or push-based. For the pull based, a *7DS* client initiates a query to the multicast group. For the push mechanism, a *7DS* client broadcasts an index of the content of its cache to the multicast group. Apart from the query client, each device may run a miniature server that consists of a cache manager, a response mechanism and/or a publishing mechanism. In the following paragraph, we describe each component.

The cache manager is in charge of organizing the local cache and searching the data. The cache size may vary for different devices. Each device maintains a cache containing information items received either from the landline network, any wide-area wireless network or other *7DS* participants. Cache items are marked up with context attributes such as display requirements, media format, general topic area such as "news" or "weather", and access restrictions. This set of attributes should be easily extendable. Queries consist of a combination of attributes and text searches.

The cache can be associated with an index file that contains the $TF * IDF$ values of a subset of terms of the files in the cache. The $TF * IDF$ value w_{ik} of a term T_k in a document D_i is defined as $w_{ik} = tf_{ik} log\frac{N}{n_k}$, where tf_{ik} is the frequency of the term T_k in document D_i, N the number of documents in the cache and n_k the number of documents that contain T_k. For the search of relevant terms, in which the client has specified several keywords $T_k,\ k = 1, \dots, m$ in his query, the receiver client will search the index-file and return the file names of D_i with the higher w_{ik}. The maximum overlapping of these files is returned. In addition, the client may send also the lines in which each word appears for first time in each file, for more hints regarding the relevance of the information.

A client queries these caches. Depending on the service or specific application, appropriate context attributes are embedded in the queries. Such attributes can be the user geographic location, application specific attributes, time or client's capabilities (e.g., display characteristics). This may take place automatically or require some user interaction. The system broadcasts the query periodically, until it receives a response.

Similarly, in a push based approach, a proxy server or a mobile host may advertise the content of its cache that are publicly accessible by sending an index of the content. The index is highly structured and it may include URLs, summary and context attributes with their values. The client receives the multicast index and filters it using some context attributes automatically or with user interaction.

As we mention, each *7DS* may act as client-querier and/or as server. It periodically checks if it receives queries. When it receives one, it may authenticate the user who sent the query. Then, it parses it to extract the context attributes and form a query to search its local cache. The result of the search is multicast. The server may include some additional information, e.g., size of a web page that corresponds to that URL, encryption methods, media format.

Both in the pull and push schemes, the client receives the collective responses sent by multiple servers. It filters the data and displays and caches them locally for browsing.

The system can set up a time interval (i.e., "collaboration time") during which it receives queries, responds to requests, advertises its index. After that the mobile switches the network interface into a low power sleep state. The mobile may alternate "sleep" and "collaborating" states. The "collaboration time" may depend on the collaboration policy and the battery level and power constraints. It can take place in a semi-automated way via a user interface in which the system displays the battery level when is below a threshold, and can also illustrate the degree of collaboration during an interval of time. Since the idling cost of the network interface is a major power consumer [5], the above policy can be used as a power saving mechanism.

2.2 Simulation Results

In this section we quantify how fast information spreads among users moving according to the random waypoint and the subway model. In addition, in the Section 2.3 we also analyze and simulate an environment where information propagates according to a simple epidemic model. We use the ns-2 simulator [6]. In all cases we measure, we compute the number of users that have the information after a period of time, the average delay that a user experiences until he or she receives the information, and also the average time until all users aquire the data. The wireless LAN is modeled as a WaveLAN network interface.

We consider a rather simple, pull-based communication protocol: When a user arrives in the system (e.g., on a platform of a station for the subway model), if it is not already a data holder, it starts periodically broadcasting a query for a data object, until it receives a response with the data. When a data holder listens to a query for data, it responds broadcasting the data. Notice that due to the broadcasting, not only the mobile who just queried for the data will receive the response, but possibly the other mobiles in close proximity.

In both of the two models, we assume that at the beginning, there is only one data holder and the rest are data queriers. The data holder is uniformly chosen from all the users. For simplicity of exposition, we assume that all the remaining users are interested in the same data item. An indicator of the data item popularity is the user interarrival time at each subway stop in the subway model, while in the random waypoint, it is the population size of the system[3].

[3]According to a study (1991), the population density of Manhattan is 4,434 people/km^2 (a total of 14 million). A user group of 25 corresponds to 0.6% population being interested in the same data item. Also, we found that the daily circulation of the NY Times (probably in the entire metropolitan area), is about 1 million. So, about 8% of the population checks the NY Times.

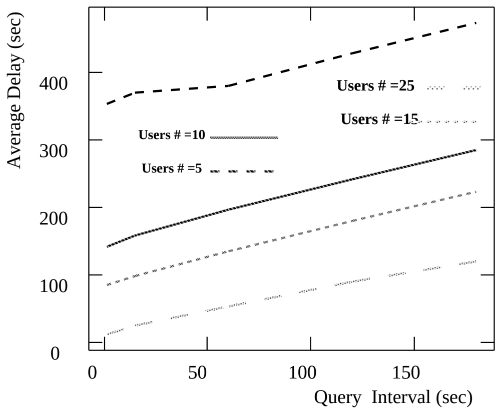

Figure 1: Average delay of a mobile to get the data for different population size and query intervals.

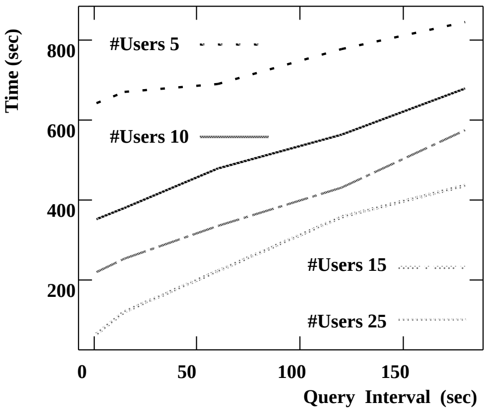

Figure 2: Average delay for the data to spread among all the users as a function of the population and the query interval.

Random Way Model

The randway model is based on Johnson's random waypoint mobility model [4]. It breaks the entire movement of a mobile host into alternating motion and rest periods. A mobile host first stays at a location for a certain time, then it moves to a new randomly chosen destination at a speed drawn uniformly from a given interval. In the random waypoint, each node i, starts from a position (x_0, y_0), and is moving towards a destination point (x_1, y_1). For each node the x_0, y_0, x_1 and y_1 are uniformly selected. Also, each node is moving to its destination with a speed u, uniformly selected from $(0\,\mathrm{m/s}, 1.5\,\mathrm{m/s})$. When a mobile reaches its destination, it pauses for a fixed amount of time *pause*, then chooses a new destination and speed (as in the previous step) and continues moving.

We fix the pause time to be 50 s and the maximum speed, 1.5 m/s. We run a set of 100 tests, each for 1500 s for every pair of population sizes $(5, 10, 25, 50)$ and query interval (1.5 s, 15 s, 60 s, 120 s, 180 s). The number of data holders at the end of each experiment is above 86%. Figure 1 illustrates the average delay for a mobile host until it becomes dataholder. As the query interval increases the average delay for the mobile host to get the data increases almost linearly. In addition, the larger is the population size the faster a user gets the data.

Figure 2 illustrates similar behavior for the maximum time for a data item to spread among the mobiles. We find that in an grid of 900 m x 900 m with 25 mobiles within 64 s the data will be propagated among all the mobiles, assuming a query interval of 1.5 s. In a setting with 25 users, the average delay is only 12 s. For a query interval of 180 s, it takes 436 s. For less popular data item, e.g., for five mobiles with a query period of 1.5 s, the propagation time (to all the users) is 643 s. In that case, the average delay for a user to acquire the data is 354 s.

We fix the query interval at 60 s. Figure 3 illustrates the average delay for the data to propagate among all the mobiles as a function of the population size in the system. As expected, as the population size increases the delay decreases. This is due to the fact that, as the population size increases, dataholders will come in close proximity with a larger number of mobile hosts and therefore the data propagation will be faster. For population size N, $N \leq 15$, the decrease is more

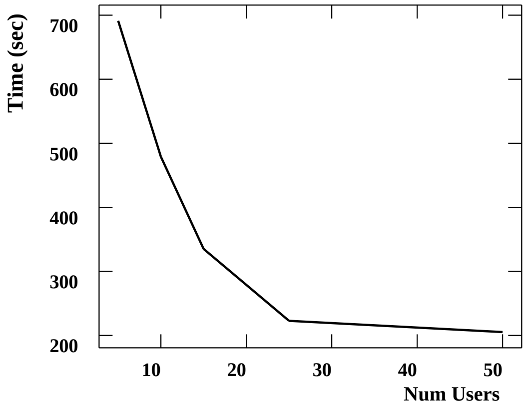

Figure 3: Average time for the data to propagate among all the mobiles as a function of the population. The querying period is 60 s.

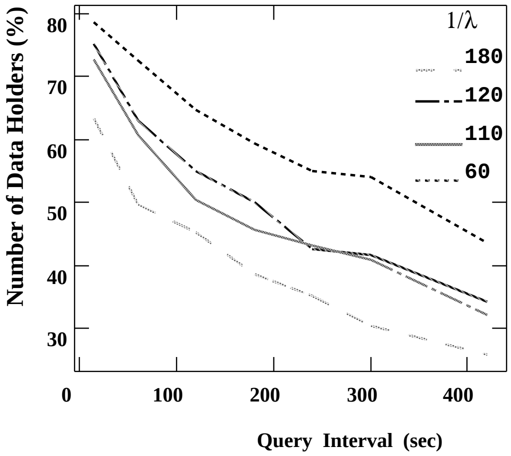

Figure 4: Percentage of data holders at the end of the observation interval for different user interarrival times ($1/\lambda$) for the subway model.

rapid, and when N reaches 25 the slope becomes smaller.

Subway Model

$1/\lambda$	Average size of the data group
60	50
110	26
120	25
180	17

Table 1: The average number of subway passengers in the data group as a function of the arrival rate $1/\lambda$.

Passengers from the kinship group arrive to subway stations as a Poisson process with a mean interarrival time of $1/\lambda$ between 60 and 180 s. A train with six cars arrives at the station every 5 minutes, stops there for 45 seconds and then travels with constant speed to the next station. The subway line has ten stops. The time between stations is uniformly distributed between 168 and 210 seconds. Passengers distribute themselves evenly across subway cars and ride the train for between 2 and 6 stations. When arriving at a station, it takes the passenger one minute to leave the platform. Passengers can exchange data with everyone else waiting on the platform or with fellow passengers in the same subway car while the train is in motion.

We investigate the influence of two parameters, namely the size of the kinship group, expressed as $1/\lambda$, and the query interval q, running 100 tests for each. We measure the spread of data for the

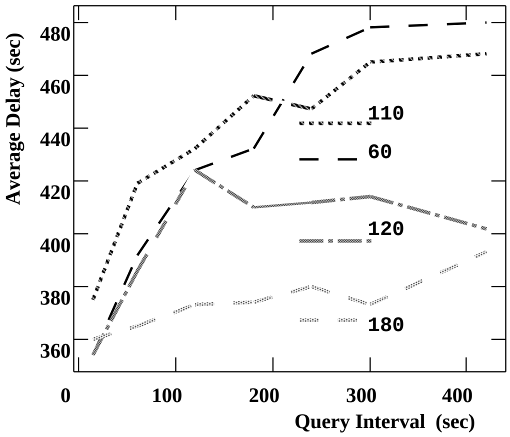

Figure 5: Average delay to get the data for the subway model for different user interarrival times ($1/\lambda$).

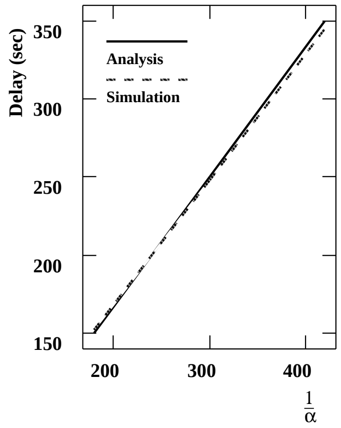

Figure 6: Delay until data spreads to all mobiles for the epidemic model (analysis and simulation).

time it takes the train to make a complete run.

The main parameters in the simulations we run are the user mean interarrival time ($1/\lambda$) and the query time interval q. We run a set of 100 tests for each ($1/\lambda$, q), where the $1/\lambda$=60, 110, 130, 180 s and q=15, 60, 120, 180, 240, 300, 420 s. The measurements correspond to an "observation interval" that starts when the train stops at the first stop and ends when it departs from the last stop, visiting a total of S train stations. We simulate a single train visiting all the stops. All the user arrivals, we consider in the observation interval, for each station, occurred during the interval of 300 s.

Figure 4 shows the percentage of users that become data holders by the end of the observation interval as a function of the query interval and the interarrival time. We observe that, as the query interval increases, the percentage of the data holders decreases. In addition, as the arrival interval decreases, the percentage of the data holders increases. This is due to the fact that the more users are around, the more likely it is for a querier to get a response. Note that in Figure 4, for query intervals below a threshold (around 240 s), the slope of the lines that correspond to higher arrival intervals is, larger, i.e., the increase in the query interval has greater effect on the number of data holders. Figure 5 illustrates the average delay that a subway passenger experiences from the time she arrives to the station till the time she receives the data. The standard deviation is quite high (in some cases it is a third or close to half the average value), and it can exceed the difference in the average delay for different interarrivals (keeping fixed the query frequency). So, it is difficult to compare the lines in the Figure 5.

2.3 Data Propagation as an Epidemic Model

The study of the data propagation among mobiles moving with various mobility patterns is not an easy problem. It is a part of a future work to study data propagation when users are moving according to a random walk. As in the previous scenarios, here, we also assume a population of N mobiles that at time 0 consists of one dataholder (the "infected" node) and $N-1$ queriers (the "susceptibles"). We suppose that in any time interval h any given dataholder will transmit data to a querier with probability $h\alpha + o(h)$. That is, with probability $h\alpha + o(h)$ any pair of mobiles gets in close proximity and if exactly one is querier, there will be data propagation exactly between the two of them. This is a simple epidemic model described in [7].

If we let $X(t)$ denote the number of data holders in the population at time t, the process $\{X(t),\ 0 \le t\}$ is a pure birth process with

$$\lambda_k = \begin{cases} (N-k)N\alpha & k = 1,..,N-1 \\ 0 & \text{otherwise} \end{cases}$$

That is, when there are k dataholders, then each of the remaining mobiles, will get the data at rate $k\alpha$. If T denote the time until the data has spread among all the mobiles, then T can be represented as

$$T = \sum_{i=1}^{N-1} T_i,$$

where T_i is the time to go from i dataholders to $i+1$. As the T_i are independent exponential random variables with respective rates $\lambda_i = (m-i)i\alpha$, $i = 1,..,m-1$, we see that

$E[T] = \frac{1}{\alpha}\sum_{i=1}^{N-1} \frac{1}{i(N-1)}$.

In Figure 6 we illustrate the expected delay for the message to be propagated in the population as a function of $\frac{1}{\alpha}$. The population size N is 5 mobiles.

3 Related Work

Scheduling algorithms for broadcasting and gossiping have been studied extensive [8, 9]. To the best of our knowledge, these studies do not consider the mobility aspect. In the area of ad hoc mobile networks the focus has been on routing protocols [10, 11, 12, 4, 13, 14]. As we mentioned in the Section 1, a difference of the traditional ad hoc networks from the mobile network we consider is that in our network the mobile hosts do not forward packets on behalf of other hosts.

There are some papers presenting mobility modeling for wireless mobile networks [15, 16]. In a cellular wireless network studies for mobility models not only aim at describing individual motion behaviors such as changes in direction and speed, but also consider the collective motion of all the mobiles relative to a geographical area (cell) over time. Models for ad hoc network [4, 15] mobility generally reflect the behavior of an individual mobile, or a group of mobiles.

The problem of the intermittent connectivity in accessing a file system has been studied. One of the earliest works, Coda [17] employs caching to enhance distributed file system availability. A selection of files are cached locally prior to disconnection. User requests are satisfied using cached data if possible, and after reconnection, cached files are propagated back. Here, we are not dealing with traditional file systems, but rather with highly structured data distributed among mobile caches.

The Bayou storage system [18] advocates a weak consistency model that aims to allow mobile client to read and write shared data. It accepts variable consistency in the desire for high availability. The database replicas eventually reach consistency. In our system the clients tolerate inconsistent data and the system may detect an inconsistency[4]. However, it does not aim in resolving it.

4 Summary and Future Work

7DS allows mobiles to quickly obtain popular data items, even if all but a very small number are disconnected from the wide-area network infrastructure. Through simulations, we were able to estimate how long a mobile user has to wait on average before obtaining a desired piece of information. This delay depends strongly on the popularity of the data and the frequency with which mobiles query for the data.

We looked at two representative mobility patterns, pedestrians moving about a city and subway riders. For pedestrians moving about randomly in a 900 by 900 m square, we found that it takes approximately a minute before everybody in a population of 25 knows if queries are sent every 1.5 s; on average, it takes only a New York minute (12 s) for a mobile to obtain the data. However, increasing the query interval by a factor of 120 to three minutes only increases the delay by a factor about seven.

Next, we modeled data distribution in a subway. 78% of the subway riders obtained the desired data at the end of their trip, assuming that other passengers with the same interests arrive once a minute. If the arrival interval increases to three minutes, the percentage of satisfied travelers drops to 63%.

Thus, in both cases, a network where users simply download information before stepping out for the morning commute can be very effective in reasonably densely populated areas, without the need to build a wireless data infrastructure.

We plan to study analytical models for some of the mobility models simulated in this paper. We are in the process of specifying the query and response protocols in more detail, implementing 7DS on laptops and PDAs running Windows CE. Currently, our prototype consists of a miniature search engine and a very small fast web proxy. The server returns URLs of the web pages in their local cache. As with typical search engines, a user may express the query with keywords.

We are also investigating the delivery of data generated by mobiles, so that those traveling in the subway can "deposit" email with their fellow passengers, to be dropped off at a network access point.

We have not presented our caching algorithms in any detail. Memory-limited mobiles will have to drop less popular items to make room for new data, particularly since much of the data we anticipate being distributed by 7DS is of a perishable nature. Also, to help with the spread of data,

[4]In cases of document sharing in collaborative applications.

mobiles may acquire items that they are not directly interested in, but have received a number of queries for earlier. This behavior is also advantageous if the system evolves to incorporate a quid-pro-quo nature, where data holders only distribute data to those who have contributed to the public good earlier.

References

[1] Bluetooth. http://www.bluetooth.com, 1999.

[2] D. L. Lough, T. K. Blankenship, and K. J. Krizman, "A short tutorial on wireless lans and ieee 802.11," *Looking forward*, vol. 5, Summer 1997.

[3] 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.

[4] J. Broch, D. Maltz, D. Johnson, Y.-C. Hu, and J. Jetcheva, "A performance comparison of multi-hop wireless ad hoc network routing protocols," in *Proceedings of the Fourth Annual ACM/IEEE International Conference on Mobile Computing and Networking (Mobicom'98)*, (Dallas, Texas), Oct. 1998.

[5] P. Gauthier, D. Harada, and M. Stemm, "Reducing power consumption for the next generation of PDAs: It's in the network interface!," research report, University of California at Berkeley, Berkeley, California, Jan. 1996.

[6] K. Fall and K. Varadhan, "ns: Notes and documentation," tech. rep., Berkeley University, 1998. Technical Report.

[7] S. M. Ross, *Stochastic Processes*. New York, New York: John Wiley and Sons, 1983.

[8] A. S. Acampora and Z. Zhang, "A throughput delay comparison; narrowband vs. broadband wireless LANs," in *Conference Record of the International Conference on Communications (ICC)*, pp. 16–22 (301.4.1–301.4.7), June 1992.

[9] S. Ramanathan and E. L. Lloyd, "Scheduling algorithms for multi-hop radio networks," in *SIGCOMM Symposium on Communications Architectures and Protocols*, (Baltimore, Maryland), pp. 211–222, ACM, Aug. 1992. *Computer Communication Review*, Volume 22, Number 4.

[10] J. Broch, D. Johnson, and D. Maltz, "The dynamic source routing protocol for mobile ad hoc networks," Internet Draft, Internet Engineering Task Force, June 1999. Work in progress.

[11] S. Corson and V. Park, "Temporally-ordered routing algorithm (TORA) version 1 functional specification," Internet Draft, Internet Engineering Task Force, Dec. 1997. Work in progress.

[12] S. Lee, W. Su, and M. Gerla, "On-demand multicast routing protocol (ODMRP) for ad-hoc networks," Internet Draft, Internet Engineering Task Force, Jan. 2000. Work in progress.

[13] Z. Haas and M. Pearlman, "The zone routing protocol (ZRP) for ad hoc networks," Internet Draft, Internet Engineering Task Force, Nov. 1997. Work in progress.

[14] S. Corson and J. Macker, "Mobile ad hoc networking (MANET): routing protocol performance issues and evaluation considerations," Internet Draft, Internet Engineering Task Force, Oct. 1998. Work in progress.

[15] X. Hong, G. P. M. Gerla, and C. Chiang, "A group mobility model for ad hoc wireless networks," in *2nd ACM International Workshop on Modeling and Simulation of Wireless and Mobile Systems (MSWiM'99)*, (Seattle), p. 8, Aug. 1999.

[16] I. Rubin and C. W. Choi, "Impact of the location area structure on the performance of signaling channels of cellular wireless networks," in *Conference Record of the International Conference on Communications (ICC)*, (Dallas, Texas), p. S56, June 1996.

[17] J. J. Kistler and M. Satyanarayanan, "Disconnected operation in the coda file system," *Proc. ACM Symposium on Operating Systems Principles*, vol. 8, pp. 213–225, Oct. 1991.

[18] D. Terry, A. J. D. M. M. Theimer, K. Petersen, M. J. Spreitzer, and C. H. Hauser, "Managing update conflicts in bayou, a weakly connected replicated storage system," in *Proceedings of the ACM Sixth Symposium on Operating Systems Principles*, ACM, Dec. 1995.