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 are intermittently connected 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 investigate how user mobility and query patterns affect data dissemination in such an environment. 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.**

I. Introduction

We envision a large percentage of the population roaming through metropolitan areas enabled with pervasive computing and communications. They will use their cellular phones, PDAs or laptops with built-in web browsers. In such environments pervasive computing devices, such as mobile user devices, sensors, Internet-connected servers, physical objects with RF-tags, will produce and store data. 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. We anticipate that mobile computing devices will be equipped with low-power, short-range (and in some cases high-speed) wireless connectivity such as BlueTooth [1] or IEEE 802.11. 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, among 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.

We describe and evaluate a system, $7DS$[1], that facilitates such information exchange for mutual benefit. 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. Napster [2] is a similar application that assists users connected to the Internet to share music files. In earlier work [3], 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. *7DS* exploits the high spatial locality of information in pervasive computing environments. Mobile users are likely to be more flexible in their information tastes, media format or quality or information accuracy.

Each device maintains a cache containing information items received by pervasive computing devices. Cache items are marked up with application specific attributes. Queries are formed using these attributes and the searches in the local caches are attributes matching.

In this paper, we focus on how user mobility patterns affect the spread of information. In particular, we investigate 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 group movement (e.g., from a platform to the train). 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 with which *7DS* participants query for data. Participants that do not have the data, query periodically till they receive it.

In the subway scenario, a train stops at a fixed number of consecutive stations. We specify a time interval with which users arrive at the stations. We measure the data propagation on the platform or in the train among the participants during that interval. By the end of the trip, for a population size of 50 users, 78% obtain the data. This population size corresponds to an interarrival time of 60 s (of one new person) at each stop. Whereas only 63% receives the data by the end of their trip,

[1] "7DS" is an acronym 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.

for a population size of 17 users [2]. Our simulations show that the ad hoc networks can be very effective in distributing popular information.

The remainder of this paper is organized as follows. Section II-A gives an overview of the infrastructure and the main components of *7DS*. Section II-B presents simulation results and Section II-C describes a simple epidemic model of data propagation with analytical and simulation results. Then, in Section III, we discuss related work. Finally, we frame our future efforts in Section IV.

II. DATA SHARING

A. Overview of the Architecture

A mobile host may also communicate with other mobiles in close proximity via a wireless LAN. We assume a well known multicast address in which the cooperating mobiles may listen and send requests and responses. The 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 publicly accessible content of its cache to the multicast group. The index is a highly structured description (summary) of the cache and may include URLs, and pairs of attribute names with their values. 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.

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 from other *7DS* participants. Cache items are marked up with application-specific attributes and support attribute-matching searches. These attributes are embedded in queries. Pull-based clients broadcast queries periodically till they receive the data. We define the *query interval* as the time that lapses between two consecutive queries sent by the same client.

As we mention, each *7DS* may act as client-querier and/or server. It periodically checks the queries it received. For each query, it extracts the attributes, performs a lookup and sends the response (if any) to the group. The server may include some additional information, e.g., size of a web page that corresponds to that URL, encryption methods, media format. To reduce the traffic at each device, clients may specify dynamically a new multicast group to receive the responses, so only interested devices receive the data (up to the application layer). Based on their profile, receivers filter and cache these indices 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 with duration that depends on the collaboration policy, battery level and power constraints [3]. 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.

B. 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 II-C we analyze and simulate an environment where information propagates according to a simple epidemic model. We use the ns-2 simulator [6] with the mobility and wireless extensions [7]. In all cases, we compute the number of users that have the information after a period of time, the average delay until the user receives the information, and the average time until all users acquire the data. The wireless LAN is modeled as a WaveLAN network interface [4].

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 until it receives it. When a data holder gets a query, it responds by broadcasting the data. Due to the broadcasting channel, not only the mobile who queried, but also the devices in close proximity will receive the data.

In both models, we assume that at the beginning, there is only one data holder and the rest are queriers. The data holder is uniformly chosen from all users. For simplicity of exposition, we fix the data object and assume that all users are interested in it.

Random Way Model

We consider nodes moving in a 900 m x 900 m grid according to random waypoint mobility model [4]. The random waypoint model 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 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 from $[0,\ 900]$. Each node is moving to its destination with a speed uniformly selected from (0 m/s, 1.5 m/s). When a mobile reaches its destination, it pauses for a fixed amount of time, 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%.

Fig. 1 illustrates the average delay for a mobile host until it

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

[3] 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.

[4] We used the WaveLAN interface simulation available with the ns-2 software.

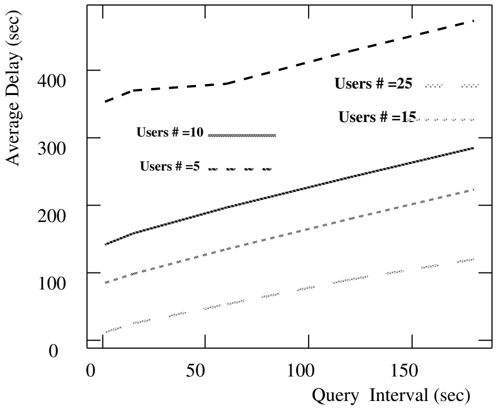

Fig. 1. Average delay of a mobile to get the data for different population size and query intervals.

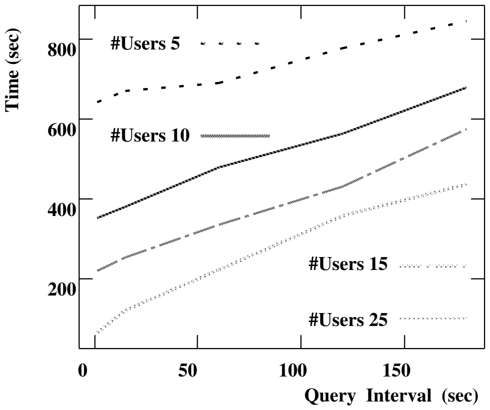

Fig. 2. Average delay for the data to spread among all users as a function of the population and the query interval.

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.

Fig. 2 illustrates similar behavior for the time data spreads among mobile users. We find that in a setting with 25 mobiles, within 64 s the data will be propagated among all the mobile users, assuming a query interval of 1.5 s. Also, 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. Fig. 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 mobiles and therefore the data propagation will be faster. For population size N, $N \leq 15$, the decrease is more rapid, and when N reaches 25 the slope becomes smaller.

Subway Model

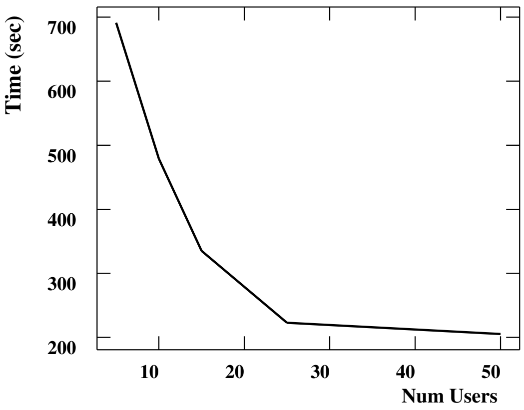

Fig. 3. Average time for the data to propagate among all users as a function of the population. The querying period is 60 s.

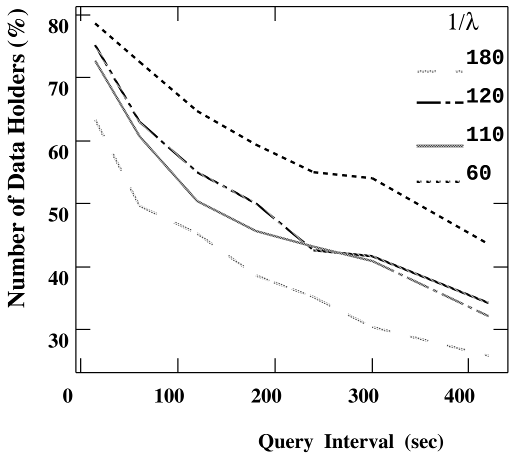

Fig. 4. Percentage of data holders at the end of the observation interval for different user interarrival times ($1/\lambda$) for the subway model.

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. The data propagation is triggered by the pull-based communication protocol we described at the beginning of the Section II-B. Here, as in the Random Way Model, the platforms and subway line layout in a grid and set their distances according to the WaveLAN wireless coverage to guarantee that only users in the same platform or car train can reach each other.

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 time it takes the train to make a complete run.

The main parameters in the simulations are the user

TABLE I
THE AVERAGE NUMBER OF SUBWAY PASSENGERS IN THE DATA GROUP AS A FUNCTION OF THE ARRIVAL RATE $1/\lambda$.

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

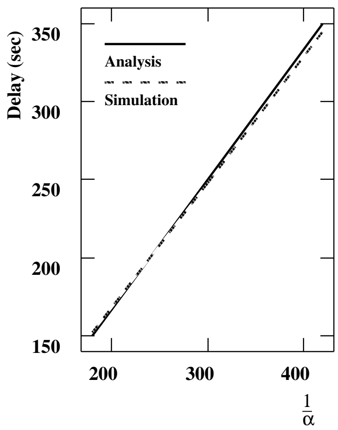

Fig. 5. Delay until data spreads to all mobiles for the epidemic model (analysis and simulation).

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 station and ends when it departs from the last, visiting a total of ten train stations. We simulate a single train visiting all the stops. All user arrivals at each station in the observation interval, occurred during the interval of 300 s.

Fig. 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.

C. Data Propagation as an Epidemic Model

The study of the data dissemination among users moving with various mobility patterns is not an easy problem [5]. As in the previous scenarios, here, we 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 [8]. 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

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

In Fig. 5 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 mobile users. In the simulations, for each pair of users, we generate events, in which, the two nodes have WaveLAN coverage (i.e., they can reach each other). The events take place during 1000 sec of simulation time, last three seconds (h) and occur with probability $h\alpha$. During these events, data holders (if any) broadcast the data.

III. RELATED WORK

Algorithms and broadcasting and gossiping protocols have been studied extensively. For example, [9] studies the problem of broadcasting in a system where the nodes are placed on a line. They present an optimal algorithm for broadcasting and compute the expected number of time steps required for it to complete. Percolation theory [10] or epidemic models have been used in the information dissemination area. When the shape theorem [10] holds for a particular setting (with respect to the nodes layout, mobility or interaction pattern), it provides elegant techniques to estimate many properties (e.g., expected time for a message to spread among all nodes). To the best of our knowledge, there is not any analytical work for the setting we described.

A protocol for information dissemination in sensor networks is [11][6]. It features meta-data negotiation prior to data

[5] In Section III, we provide some references for this problem.

[6] The sensors are fixed.

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.

In the area of ad hoc mobile networks the focus has been on routing protocols [12], [13], [4]. 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 a few papers presenting mobility modeling for wireless mobile networks, such as [14], [15].

IV. Summary and Future Work

$7DS$ allows mobile users to quickly obtain popular data items, even if most users are disconnected from the wide-area network infrastructure. Through simulations, we 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 grid, we found that it takes approximately a minute before everybody in a population of 25 obtains the data item, if queries are sent every 1.5 s; on average, it takes only 12 s for a mobile user 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%.

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 a query with keywords. We would like to investigate other designs that incorporate prefetching mechanisms, in which $7DS$ "guesses" what data to query on behalf of the user. One facet of the project is to experiment with the system and understand characteristics and constraints of this type of applications.

Memory-limited mobile users 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.

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

As an alternative to the infrastructureless environment, we plan to investigate how distributed Internet-connected hosts attached to data repositories can assist *7DS* clients to locate the relevant data. The repositories store data generated by the pervasive computing devices. They communicate with other repositories to learn about their contents and act as application-based routers to route queries. $7DS$ clients query their local repository (via a short-range wireless network). We plan to investigate the deployment of such an infrastructure with caching schemes that exploit some of the fundamental aspects of multi-sensor systems.

References

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

[2] "Napster." http://www.napster.com/, 2000.

[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] "Wireless and mobility extensions to ns-2." http://www.monarch.cs.cmu.edu/cmu-ns.html.

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

[9] K. Ravishankar and S. Singh, "Broadcasting on [0,L]," *Discreet Applied Mathematics*, 1994.

[10] R. Durrett, *Lecture notes on particle systems and percolation*. Pacific Grove, CA, 1988.

[11] 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.

[12] 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.

[13] 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.

[14] X. Hong, M. Gerla, G. Pei, 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.

[15] 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.