Usenet-on-the-fly - supporting locality of information in spontaneous networking environments

Christian Becker, Martin Bauer[1], Jörg Hähner[2]
University of Stuttgart, IPVS,
Breitwiesenstr. 20-22
70565 Stuttgart, Germany
+49 711 7816 (357 | 275 | 218)
{becker | bauer | haehner}@informatik.uni-stuttgart.de

ABSTRACT

People on the move are typically interested in information with respect to their proximity. Location-based services in general supply users with information about their proximity typically relying on an infrastructure storing the information and tracking the mobile objects, i.e. users. In this paper we present an approach for spontaneous, i.e. ad hoc, networks inspired by the Usenet. Information is exchanged using a peer-to-peer synchronization mechanism. The information is made available through channels grouping related information. The information propagation is solely based on spontaneously connected devices not requiring any infrastructure. Our prototype implementation shows the technical feasibility of our approach, whereas simulation results show the applicability of information diffusion in outdoor scenarios with a realistic number of nodes, covering a city center.

Keywords

Ad hoc networks, information dissemination, diffusion

INTRODUCTION

Location-based services (LBS) gain popularity. While many commercial approaches are tied to the cellular phone infrastructure, e.g. [5][14][20], researchers address solutions for the indoor domain [6][11][23] and outdoor domain [3][16] or both [7] based on their own infrastructure. Common to these approaches is the necessity of an infrastructure storing location-dependent data and management of user positions.

The availability of small computing devices, e.g. Personal Digital Assistants (PDAs) or cellular phones, equipped with short range radio transmission technologies such as Bluetooth or IEEE 802.11 allows information exchange on a peer-to-peer basis whenever two devices are within each other's radio range. Additionally, information of the environment can be captured from sensors equipped with similar radio technology.

The overall focus of our research is to investigate what mechanisms are needed to support applications for mobile users in a ubiquitous computing environment using ad hoc communication. The goal is to provide a foundation that allows users to successfully interact with other users and their environment. This includes collecting and providing information about the spatial context of the user.

A major problem in mobile ad hoc networks is the management and dissemination of information. Since the mobile devices are restricted in their resources, a complete replication of information will not be possible. Information exchange should be restricted with respect to the spatial scope of the information and the interests of the user. Another issue is the multitude of available information: how can a user determine or specify which information is interesting to him or her?

In this paper, we present an approach for information dissemination based on epidemic algorithms, i.e. diffusion. As one possible scenario, imagine that it is Saturday night and a large number of young people are walking around in the city center looking for some fun. What they really want to know is what is currently going on: where are the cool parties, the hip discos or the most popular bars. So the information needed depends very much on the current context of the user, especially the location. The distribution of the information can be asynchronous and possibly anonymous, which fits well with our proposed diffusion-based approach.

If cellular phones are equipped with short range radio technologies, such a Bluetooth, they are the ideal devices for our scenario. Almost all young people have cellular phones and use them frequently for writing SMS messages, especially in Europe. (The SMS Service allows the transmission of short text messages between cellular phones using the cellular phone infrastructure). So the general technology is well-introduced, the only difference being that the messages are exchanged using ad hoc connections between devices in the proximity.

[1] Martin Bauer is funded by Microsoft Research, Cambridge, UK

[2] Jörg Hähner is funded by the Gottlieb Daimler and Karl Benz Foundation, Ladenburg, Germany

In order to structure the information exchanged between devices, the information is grouped into channels according to subjects, similar to newsgroups in the Usenet. Users can subscribe and unsubscribe to channels. Information is only propagated in a distinct area with respect to its locality. As a result, an easy-to-use application can provide users with information about their proximity.

To show the technical feasibility of our concept, we have built a prototype application. However, since the usefulness of our application can only be determined based on a large user population and since the technology is not yet widespread enough for a large-scale usability study, we have conducted a number of simulations to provide some evidence that the diffusion-based approach makes sense in the given context. An important aspect of the simulation is the mobility of the users. Therefore, we need a mobility model reflecting the characteristics of user mobility that may have an influence on the diffusion. In this paper we take two mobility models and compare the results of the respective simulations: the random waypoint model that is widely used for the evaluation of algorithms in ad hoc networks and a graph-based mobility model [21] that takes the possible user paths, i.e. streets, into account and is therefore more realistic.

The structure of our paper is as follows: In the next section we present our general system model. Then we describe our application scenario, focussing on the "Usenet-on-the-fly" prototype, followed by a detailed description of the underlying information dissemination protocol. After that we present simulations of the information dissemination protocol and discuss their results. Following a discussion of related work, the paper concludes with an outlook on future work and a summary.

SYSTEM MODEL

The system consists of mobile nodes users carry. Examples of such nodes are devices like cellular phones or PDAs capable of short range radio transmission. The communication between nodes occurs spontaneously, i.e. whenever two devices are within radio range of each other, they discover each other and can exchange information. Additionally, sensors or info stations may provide local information of the environment to the thereby formed mobile ad hoc network (MANET).

The information exchanged in such a MANET can differ widely and is obviously application-dependent. For the remainder of this paper we consider information to be of local interest. Dishes of the day, temperature of rooms, bus schedules are mostly relevant in the proximity of their real-life source. Since we do not assume any access to an infrastructure we want to investigate, how peer-to-peer computing in such spontaneously formed networks can be used for information dissemination.

We assume the nature of information to be "nice to have". If some information was critical to a user, the user would pay for an uplink to an infrastructure, e.g. via wireless cell-based communication.

APPLICATION SCENARIO

The application scenario we want to look at is concerned with the propagation of information with a local scope in a MANET. We do not consider multi-hop messages, e.g. routing, here, but only dissemination of information with multiple, previously unknown receivers. The information, as mentioned before, is assumed to be locally relevant. Hence, an information dissemination protocol has to discard the information when the scope of the information is left. Recipients of information must be provided with a classification of the information in order to decide, if they want to accept it and store it locally.

Due to the multitude of information and corresponding information possible in such scenarios, we reduce the complexity by focussing on a simple scenario. The aim is to provide an evaluation of diffusion-based information dissemination and demonstrate how information with local relevance can be handled in such ad hoc scenarios.

The information in the context of this paper is represented as a message. A message contains a source, which created the message, a topic, which classifies the content of a message, and a body carrying the information of the message.

Messages could represent sensor data, with the sensor ID as source, the kind of sensor information, e.g. temperature or humidity as topic, and the currently sensed value as content of the message. Another example could be the provision of bus schedules, where the distinct bus station is the information source and the topic would determine a transport schedule with the message body containing the next bus departure. Moreover, users could also provide information, e.g. rankings of restaurants or shop offers and feed them into the system by creating messages. Actually, this inspired our prototype application - Usenet-on-the-fly - which is presented in the next section. Following that, we explain the information dissemination protocol in more detail and present some simulation results.

THE USENET-ON-THE-FLY PROTOTYPE

The Usenet provides users with the ability to subscribe to so-called newsgroups where they can read, post and reply to articles. The newsgroups group articles with a distinct topic. It is considered rude in the Usenet community to place articles in inappropriate groups - being "off-topic". The Usenet does not rely on a centralized infrastructure. Instead, servers providing "news" to users allow them to read, post and reply to articles. This local news is propagated over news feeds to other news servers which present these articles to their users, receive the replies and postings and offer these as news feeds to other news servers.

The architecture of the Usenet originates from former times when many computers were not permanently linked to each other as nowadays via the Internet. However, this situation reflects the characteristics of an ad hoc network where nodes are not permanently available but only when they are in the vicinity of other nodes. The concept of categorizing information by grouping them into newsgroups according to topics - or in our terminology: channels - and peer-to-peer reconciliation of content matches the needs of information propagation according to our requirements.

The data model of our Usenet-on-the-fly is directly corresponding to the messages as they were informally defined in the previous section:

- Message headers, i.e. the channel name, the subject and the sender
- Message content, i.e. the actual information

Scoping of the information is simply done by adding a hop count. Thereby, the scope within which a message is presented to other nodes is restricted, which, in most cases, automatically leads to a geographical scoping.

Functionality

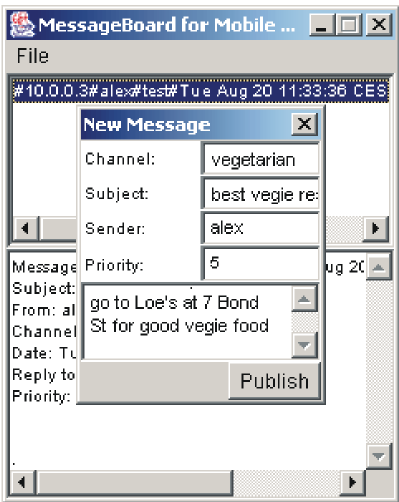

Figure 1: Usenet-on-the-fly User Interface

Figure 1 shows the user interface of the Usenet-on-the-fly prototype. The prototype is realized as a Java application. We used notebooks and Compaq iPaqs equipped with WaveLan cards as an evaluation platform.

A user can create channels and messages and open an existing channel to retrieve messages. Figure 1 shows the dialog for creating a message. This dialog combines the creation of a new message with the possible creation of a new channel. Additional attributes, e.g. the priority, allow filtering of messages in order to save bandwidth or space on the devices.

Users can subscribe to a topic, i.e. a channel, and receive all messages on that channel. The local database containing the messages is updated whenever another node is met. Both nodes negotiate about their channels and contents and exchange the difference. New channels are presented to the user who can subscribe to them or simply ignore them.

When users on the move are visiting different places, only the information concerning these places is offered in the channel. Information is scoped in its lifetime by a time-to-live (TTL) as well as in its propagation scope by a hop count. The message exchange is based on a single hop communication, i.e. devices only communicate with other devices in their transmission range. Hence, restricting the number of times a message can be passed on between nodes leads to a geographical scoping.

As an example consider a user subscribing to “restaurant menu”, “bus schedule”, and “restaurant recommendations”. The channel “restaurant menu” will contain the dish of the day of the restaurants within a distinct vicinity, depending on the hop count. Also, only the bus schedules of nearby bus stops are presented in the “bus schedule” due to the scoping. Not only stationary entities like restaurants or bus stops can create messages. Other users can use the “restaurant recommendation” channel to express their satisfaction about a particular restaurant. This information is scoped with respect to its local lifetime and geographical scope as well.

Architecture

The Usenet-on-the-fly prototype was built in a straightforward way. Nodes maintain a small database where the channels and all messages are stored. The database is regularly scanned and messages whose TTL has expired are deleted. The user interface operates on the database and allows the display of channels and their messages as well as the creation of new messages and replies.

The content of local databases are synchronized with other nodes whenever they are within their radio transmission range. First, the channels are compared and new channels and messages are announced to other nodes. Before offering a message, the hop count is considered. If the scope of a message has been reached, it is no longer propagated to other nodes.

The resulting architecture is depicted in Figure 2. Central to the system is the database where messages are stored in the corresponding channels. The user interface accesses the database in order to display available channels and on selection of channels the messages of the channel. The user can create new messages and channels leading to new data in the database.

The content of the database is propagated by a simple diffusion protocol. We will describe the protocol in detail in the next section. For short: the protocol announces locally available data to other nodes. These nodes can request the information and store it in their databases. After a node has propagated its database content, it switches the role and updates its database by the advertised channels and messages of the other node. This data reconciliation occurs whenever two nodes “meet”. To allow nodes to continue exchanging data when they stay in communication range, they can end their communication and then “rediscover“ each other.

The communication subsystem is built on top of a minimized servlet container which offers the Simple Object Access Protocol (SOAP) for message exchange. The SOAP standard is well-suited for interoperability between different platforms. However, it currently restricts us to unicast communication, so we cannot take advantage of broadcast protocols that are suitable for propagating data to a larger set of recipients.

The prototype has been built in Java. The platforms for evaluation were notebooks and Compaq iPaqs, both equipped with WaveLan. It is available for download from http://www.informatik.uni-stuttgart.de/ipvr/vs/de/people/haehnejg/#misc

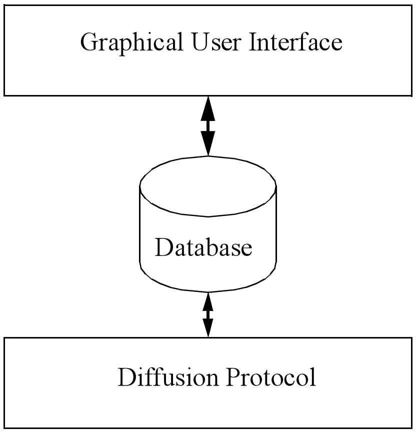

Figure 2: Usenet-on-the-fly Architecture

INFORMATION DISSEMINATION PROTOCOL

The messages in the system are disseminated using a diffusion-based protocol that we call Channel and Message Diffusion Protocol with Negotiation (CMDPN). The pseudo-code of an algorithm implementing that protocol on a given node is shown in Figure 3.

When a node A discovers another node B in its transmission range, it sends an advertisement message listing all the channels (consisting of a unique channel ID and a description of the channel topic) and the IDs of the messages it currently has in its database. Node B then goes through the advertised channels and checks, if it has seen them before. If not, the user is given the channel description and is asked, if he or she wants to subscribe to the new channel. Having updated the subscription information, Node B goes through the advertised message IDs pertaining to those channels it has subscribed to. It creates a request message containing the message IDs of the messages it does not have in its database yet. On receiving the request from Node B, Node A collects the requested messages and sends them to Node B, which updates its database accordingly.

```
TYPES
message_id: unique id
channel_id: unique id
topic: string
message_body: string

message: struct
    channel_id
    message_id
    message_body

VARIABLES
channel_topic = array[channel_id] of topic
message_ids = array[channel_id] of
                    list of message_id
messages = array[message_id] of message
seen_channels = list of channel_id
subscribed_channels = list of channel_id

EVENT HANDLERS
ON_NODE_DISCOVER()
 Channel_ADV ca = empty list
 Message_ADV ma = empty list
 for each channel_id in subscribed_channels
                                        do
    append(ca, (channel_id,
         channel_topic[channel_id]))
    for each message_id in
         message_ids[channel_id] do
       append(ma,
         (channel_id , message_id))
    od
 od
 if not empty(ma) then
    send_message((ca, ma))

ON_RECEIVE_ADV((ca: Channel_ADV,
              ma: Message_ADV)):
 for each (channel_id, channel_topic) in ca
                                       do
    if channel_id not in seen_channels then
       append(seen_channels,
                 channel.channel_id)
       if ask_user(channel_topic) then
          append(subscribed_channels, channel)
          channel_topic[channel_id]:=
                             channel_topic
       fi
    fi
 od
 Message_REQ mr = empty
 for each (channel_id, message_id) in ma do
    if channel_id in subscribed_channels then
       if needed(message_id) then
          append(mr, message_id)
 od
 if not empty(mr) then
    send_message(mr)

ON_RECEIVE_REQ(mr: Message_REQ):
 Message_DATA md = empty
 for each message_id in mr do
    append(md, messages[message_id]))
 od
 if not empty(md) then send_message(md)

ON_RECEIVE_DATA(md: Message_DATA):
 for each message in md do
    if needed (message.message_id) then
       append(message_ids[message.channel_id],
             message_id)
       messages[message_id]:= message
    fi
 od
```

Figure 3: Channel and Message Diffusion Algorithm with Negotiation (CMDPN)

Of course, the same protocol is applied in the other direction between node B and node A

The CMDPN is a simple protocol for replicating Usenet-style messages. Its purpose is to minimize the exchange of unwanted messages, saving bandwidth and energy, which are scarce resources for mobile devices using wireless connections.

Further improvements could be:

- To further reduce the data that needs to be exchanged, the protocol could be split up into two phases. In the first phase only the channel information is exchanged. Then, in the second phase, only the information about the messages pertaining to those channels a node is subscribed to need to be exchanged.
- The actual messages could be exchanged according to user-defined priorities, which is especially helpful, if the devices are not within communication range long enough to exchange all messages of interest.
- A history of mobile nodes and the data which has recently been exchanged with them could be kept to keep the message advertisement messages small.

In the following section, we will present some simulations to evaluate the effectiveness of the dissemination of messages to a population of nodes, i.e. how many nodes have received a certain message in what period of time.

SIMULATIONS

This section will describe the simulation context used for the evaluation of the CMDPN protocol, as well as the results, including a discussion of the results.

Since, as a first step, we were mainly interested in the maximum effectiveness of message dissemination, given a certain population of nodes, we used simplified assumptions:

We assume that all nodes are interested in all the messages and that these messages pertain to a single channel. We did neither restrict the hop count, nor set a restrictive TTL, so the messages are distributed within the whole area over the time of the simulation. To control the introduction of messages into our system, so that we could more easily calculate the spreading of information, we assume that the messages are introduced into the system by special stationary „sensor nodes“, e.g. providing the local temperature or introducing the meal of the day. Each of the sensor nodes continuously provides the same single message to the mobile nodes in its proximity.

The CMDPN protocol was simulated using our Java-based CanuSim simulator, which implements a simple MAC layer that prevents multiple nodes from accessing the same wireless channel simultaneously. The main advantage of our simulator is that the two mobility models we present in the following can easily be integrated.

Simulation Model

The simulations were performed for an outdoor context using two different mobility models. The first model is the so-called random waypoint model (RWP). This model is often used for the evaluation of algorithms in the area of MANETs [2] and originated in the application area of rescue and disaster operations. In the RWP model a mobile node chooses a random destination and a speed and then moves directly to the destination using the given speed. The size of the area covered was 2462 m x 1733 m, equivalent to the area of the city center used later on.

The second mobility model - the graph-based mobility model GBM [21][19] - assumes that mobile nodes do not move randomly, but according to an infrastructure, e.g. road map or building layout. It models the spatial environment as a graph. The example graph for our simulations models a typical city center, as it can be found in Central Europe. The model contains 115 locations on an area of 2462 m x 1733 m interconnected with 150 edges. Figure 4 shows a sketch of the city graph used.

In several scenarios different numbers of mobile nodes as well as sensor nodes, each providing one piece of information, were placed randomly on the graph. Destinations were chosen randomly out of the 115 locations in the graph scenario or randomly in the RWP scenario. The mobile nodes moved around at normal pedestrian speed, i.e. between 3 and 5 km/h.

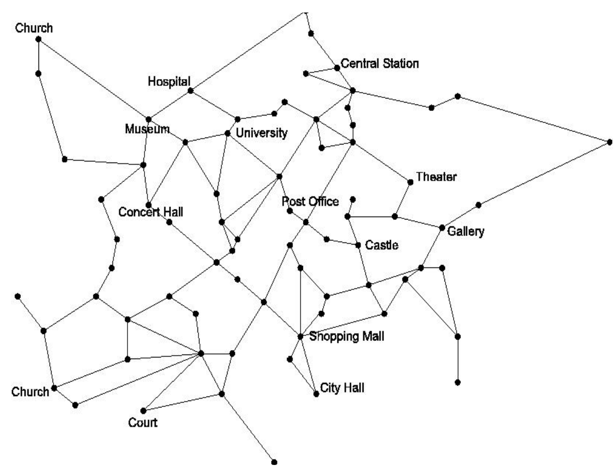

Figure 4: City Center Graph

On reaching a destination, mobile nodes stayed there for 12 to 20 minutes, representing pedestrians stopping at a shop or station, before choosing a new random destination. The sensor nodes remained stationary, broadcasting their sensor information to mobile nodes within transmission range. The mobile nodes all used CMDPN as the protocol for message exchange. The time needed to discover a node in transmission range was assumed to be between 2 and 3 seconds, which corresponds, for example, to average Bluetooth discovery times [10]. All scenarios used a transmission range of 75m.

The simulation runs were terminated when a certain level of *information spreading* was reached. The information spreading is calculated by

$$is = \frac{\sum_{m \in M} \mathrm{databasesize}(m)}{|M| \times |S|}$$

where M denotes the set of mobile nodes and S denotes the set of sensor nodes that each introduced a single message (500 bytes) into the system. The function *databasesize(m)* sums up all the messages stored on the mobile node *m*.

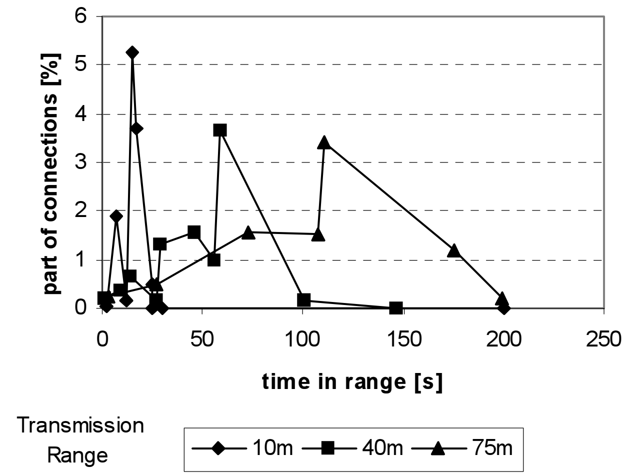

Figure 5: Time in Communication Range Depending on the Transmission Range between Mobile Nodes and Stationary Sensor Node

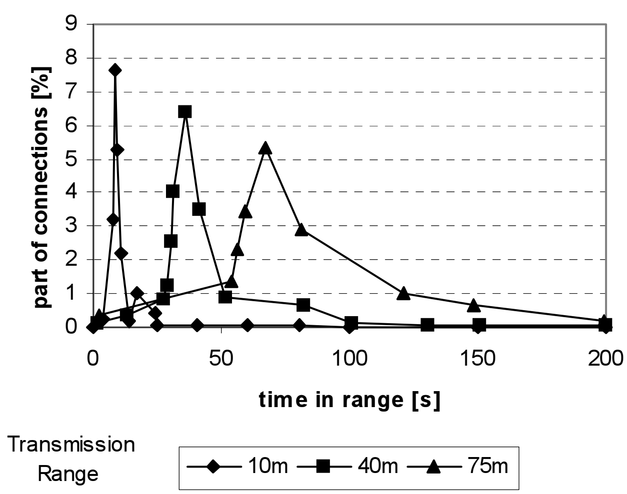

Figure 6: Time in Communication Range Depending on the Transmission Range, between Mobile Nodes

We have integrated the graph shown in Figure 4 into our simulator for MANETs in order to simulate realistic mobility patterns of users. A first interesting result concerning the relation between communication time and transmission range is shown in Figure 5 and Figure 6. We placed 100 sensor nodes in the city center scenario and measured the average time of communication between 1000 mobile nodes and the 100 sensor nodes. The figures are based on a one hour simulation of the scenario.

Figure 5 shows the distribution of time mobile nodes are in communication range with any sensor node. Assuming a communication range of 75 meters most nodes have more than 50s per connection to communicate, whereas 10m transmission range allows only less than 15s for most connections.

Figure 6 shows the distribution of communication time between the mobile nodes. Here the results are slightly worse for the 10m transmission range, since the mobility of nodes shortens the transmission time to 7 seconds, whereas for 75m most nodes still have more than 50s per meeting. Information dissemination in such ad hoc networks has to be aware of these small slots for communication, i.e. not relying on stable routes and long-term communication relations

Simulation Results

This subsection presents the simulation results of the aforementioned diffusion algorithm based on random waypoint versus the graph-based mobility model.

Figure 7 and Figure 8 present the simulation results for the RWP scenario based on a transmission range of 75m. The simulations were run until 95% information spreading was reached, meaning that every mobile node carried almost all the information disseminated by the sensor nodes. The results show that a higher number of mobile nodes supports the information spreading. Nevertheless even a small number of nodes leads to a reasonably fast message replication considering the large area: 100 mobile nodes discovered 100 sensor nodes on an area of approximately 4 square kilometers reaching an information spreading of 95% in only little more than two hours moving at pedestrian speed. A larger number of mobile nodes (500) reaches the same amount of information spreading in less than half an hour. Remember that 500 people in a city center is still a fairly small number.

Figure 9 shows the results of the simulations performed with the GBM pattern. The obtained results show a significant improvement over the results of the RWP pattern. The graph-based simulations showed to be approximately twice as fast until the information spreading of 95% was reached. The major reason for this improvement is the fact that the mobile nodes only move along the edges of the graph and do not occupy the whole area as they do in the RWP model. Since the GBM pattern represents our initial outdoor scenario better, we expect CMDPN to behave towards those results in a "real world deployment".

A second set of simulations with only one sensor node was conducted to investigate the effect of the spreading of a single information item. Figure 8 and Figure 10 show the results obtained with RWP and GBM pattern respectively. The results show that, once the information has been picked up and passed on a few times, the steepest rise of the curve is reached. This shows that the information is spread very quickly around its source supporting the locality aspect of many information items in ubiquitous computing.

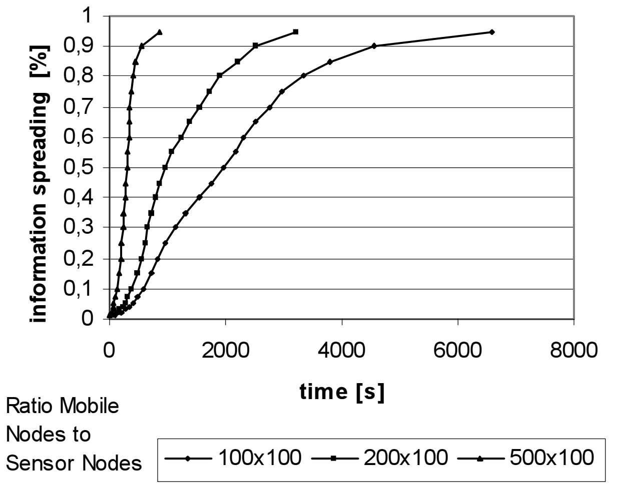

Figure 7: Information Spreading over Time for Random Waypoint Movement with 100 Sensor Nodes

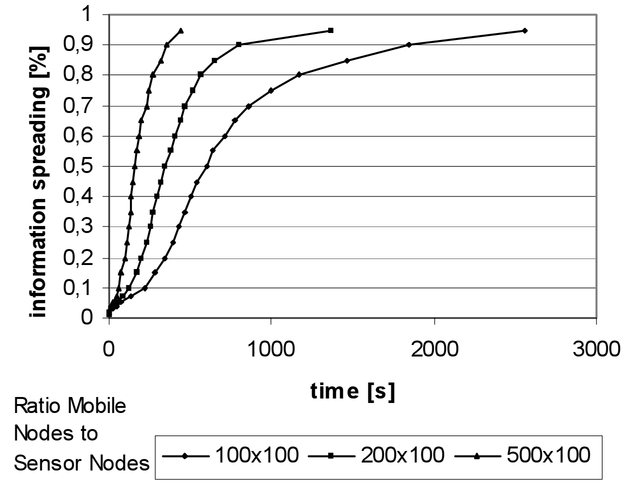

Figure 9: Information Spreading over Time for Graph Walk Movement with 100 Sensor Nodes

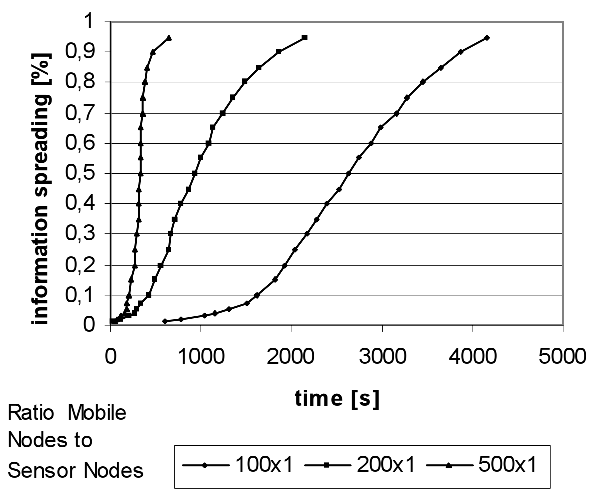

Figure 8: Information Spreading over Time for Random Waypoint Movement with One Sensor Node

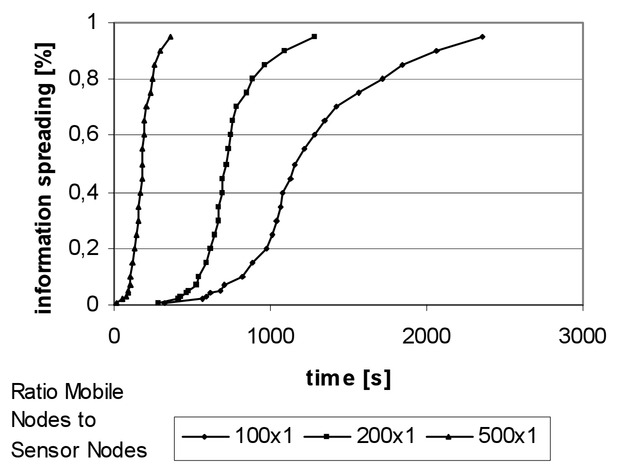

Figure 10: Information Spreading over Time for Graph Walk Movement with One Sensor Node

Discussion

The simulation results show that, assuming realistic values for the transmission range and the density of mobile nodes, the distribution of an update in the vicinity of an information source is a matter of minutes. Reaching an almost complete spreading of information can also be achieved within less than an hour.

The difference between the simulation results based on different mobility models shows the importance of using realistic mobility models in order to get realistic simulation results.

RELATED WORK

Applications based on message exchange in infrastructure-based systems such as the Usenet have been used for a very long time. More recently, peer-to-peer file sharing application like Gnutella [9] have become popular. However, as we are interested in systems based on ad hoc networks with mobile nodes, we want to mostly restrict our discussion in the following on systems fitting those characteristics.

In the area of collaborative wearable computing, the ad-hoc exchange of information between mobile users during chance encounters has been investigated. Application scenarios including the exchange of tasks between user agents [12] and the dissemination of trust information [18] have been simulated. In those scenarios the information distributed is much more specialized than in ours and exchange of information depends much more on the individual users themselves, even though the underlying mechanisms are very similar.

Much work has been done on routing in MANETs, where messages between sender and receiver are exchanged on [4] an unstable path built of mobile nodes (see, for example, [17] or [8]). Usually, however, a fully connected path from the sender to the receiver is required to be able to forward a message. In [22] a general routing protocol for partially connected networks is discussed, which, similar to our approach, uses the moving mobile nodes to relay messages. Their results show, that such an approach is feasible, transmitting 100% of the messages in most cases in reasonable time.

Recently, information diffusion has been discussed in the area of sensor networks. There, information is exchanged between a number of randomly placed non-mobile nodes, which acquire a model of their environment using built-in sensing systems (e.g., for seismic data or images). Algorithms in this area have to be able to cope with the failure of single sensors. Different variants of broadcast algorithms have been discussed for such sensor networks with the goal of reducing bandwidth and energy consumption [25]. In [13] a family of negotiation-based protocols for sensor networks, called SPIN, are discussed. It is shown that they perform better regarding performance and energy consumption than the more simple broadcast protocols.

More closely related to our approach regarding the dissemination of data is the 7DS system [15]. Their underlying data model relies on a hierarchy of web-caches and the information can be accessed via a client/server-based approach from an infrastructure if available. If network partitions occur, the mobile nodes rely on their cached data, which can be updated similar to the diffusion algorithm that we have presented here. Cooperation among the mobile nodes allows the access of information in other caches. Queries trigger a diffusion process of data through the mobile nodes which update their cache with the requested information. The mobility model is a strict random waypoint model neglecting spatial constraints. However, their objective relates to the fragmented data storage. Replication of the data on every node and restricting the coverage of the information dissemination, e.g. to an area or a number of nodes, is not an issue there.

CONCLUSION AND OUTLOOK

In this paper we have shown that the dissemination of data in large MANETs is feasible. Simulation results show that information can be spread among several hundred users in a city center scenario within approximately 10 to 60 minutes from its initial creation at the sources. This time interval is appropriate for many types of information that may be of interest to a pedestrian walking through a city, such as information about current events or specials on sale. A *1-to-many* message exchange application, like our Usenet-style prototype for PDAs, can be implemented using devices and technologies which will soon be deployed among many (millions of) users.

It can be concluded that information can be made available in MANETs through the dissemination of messages using a diffusion algorithm. This means that in a lot of cases it is not necessary to access an infrastructure in order to obtain information concerning the current proximity.

After showing the general feasibility, there are still many questions that remain unanswered. Our simulations show that the time it takes to disseminate information among users varies depending on how the mobile nodes, i.e. the users, move around. This strengthens the need for realistic user mobility models in order to obtain a reliable performance prediction of new systems prior to deployment.

More simulations are needed to show, if limiting the scope of the information by hop counts successfully approximates the locality of information as we expect. In any case, this solution gives only a very coarse resolution of locality. A more sophisticated solution here is to employ a more detailed world model [1] and location sensors, e.g. GPS, to locate the mobile nodes. Then, the spatial scope of an information could be specified directly and precisely, e.g. information could be interesting for users on the same floor only, but not for those on the floor below, yet the people there might be closer concerning the communication if only distance is taken into account.

In our simulations we have assumed that every user is interested in all channels, i.e. we completely replicated all messages. If we assume a large variety of topics for channels in a real-world system, it is evident that complete replication does not work due to resource restricted devices on the one hand and users that are not willing to carry unwanted information on their devices on the other. A technical compromise is that every user allots some portion of his systems memory and communication time for information that is not of (high) interest tohim. The question here is: how much is technically necessary for the system to work and how much are users willing to contribute.

Since our system is based on the assumption that the number of participants is large, the social situations in which a user sees a clear benefit and therefore uses this interaction style has to be investigated. This, we believe, can only be answered by appropriate user studies.

Acknowledgements

We want to thank our students Alexander Rau and Mario Neynens for their work on the Usenet-on-the-fly prototype. Special thanks also to Illya Stepanov for his work on the implementation of the CanuSim simulator.

REFERENCES

1. Bauer, M., Becker, C., and Rothermel, K. Location Models from the Perspective of Context-Aware Applications and Mobile Ad Hoc Networks. In: *Workshop on Location Modeling for Ubiquitous Computing*, UBICOMP 2001, Atlanta, 2001.
2. Broch, J., Maltz, D., Johnson, D., Hu, Y.-C., and Jetcheva, J. A performance comparison of multihop wireless ad hoc network routing protocols. In *Proceedings of the Fourth International Conference on Mobile Computing and Networking (MobiCom)*, Dallas, Texas, 1998

3. Cheverst, K., Davies, N., Mitchell, K., Friday, A., and Efstratiou, C. Developing a Context-aware Electronic Tourist Guide: Some Issues and Experiences, In *Proceedings of CHI 2000*, Netherlands, April 2000

4. Demers, A., Greene, D., Hauser, C., Irish, W., Larson, J., Shenker, S., Sturgis, H., Swinehart, D., and Terry, D.: Epidemic Algorithms for Replicated Database Maintenace. In *Proceedings of the 6th ACM Symposium on Principles of Distributed Computing*, pp. 1-12, 1987.

5. GSM World. Location-based Services. http://www.gsmworld.com/technology/applications/location.shtml

6. Harter, A. Hopper, A., Steggles, P., Ward, A., and Webster, P. The Anatomy of a Context-Aware Application, *Proceedings of the Fifth International Conference on Mobile Computing and Networking (MobiCom'99),* Seattle, WA, USA, pp. 59-68, 1999.

7. Hohl, F., Kubach, U., Leonhardi, A., Rothermel, K., and Schwehm, M. Next Century Challenges: Nexus - An Open Global Infrastructure for Spatial-Aware Applications. In *Proceedings of the Fifth Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom'99)*, Seattle, WA, USA, pp. 249-255, 1999.

8. Johnson, D. Routing in Ad Hoc Networks of Mobile Hosts. In *Proceedings of the Workshop on Mobile Computing Systems and Applications*, Santa Cruz, CA, USA, 1994.

9. Kan, G. Gnutella. In *Peer-to-Peer: Harnessing the Power of Disruptive Technologies*, Andy Oram (ed.), March 2001.

10. Kasten, O., and Langheinrich, M. First Experience with Bluetooth in the Smart-Its Distributed Sensor Networks. Workshop on Ubiquitous Computing and Communication. In *Proceedings of 10th International Conference on Parallel Architectures and Compilation Techniques (PACT'01)*, Barcelona Spain, 2001.

11. Kidd, C., Orr, R., Abowd, G., Atkeson, C., Essa, I., MacIntyre, B., Mynatt, E., Starner, T., and Newstetter, W. The Aware Home: A Living Laboratory for Ubiquitous Computing Research. In *Proceedings of the Second International Workshop on Cooperative Buildings* - CoBuild'99, 1999

12. Kortuem, G., Schneider, J., Suruda, J., Fickas, S., Segall, Z. When Cyborgs Meet: Building Communities of Cooperating Wearable Agents. In *Proceedings Third International Symposium on Wearable Computers (ISWC'99)*, San Francisco, CA, USA, 1999.

13. Kulik, J., Heinzelman, W., and Balakrishnan, H. Negotiation-based protocols for disseminating information in wireless sensor networks. In *Proceedings of the Fifth Annual ACM/IEEE International Conference on Mobile Computing and Networking*, Seattle, WA, USA, 1999.

14. LBS Portal, http://www.lbsportal.com/

15. Papadopuli, M., and Schulzrinne, H. Effects of power conservation, wireless coverage and cooperation on data dissemination among mobile devices. In *Proceedings of MobiHoc 2001*, Long Beach, USA, 2001.

16. Pascoe, J. The Stick-e Note Architecture: Extending the Interface Beyond the User. In *Proceedings of the International Conference on Intelligent User Interfaces*. Editors, Moore, J., Edmonds, E., and Puerta, A., pp. 261-264, 1997

17. Perkins, C.E., and Bhagwat, P. Highly Dynamic Destination-Sequenced Distance Vector Routing (DSDV) for Mobile Computers. In *Proceedings of ACM SIGCOMM'94*, London, UK, pp. 234-244, 1994.

18. Schneider, J., Kortuem, G., Jager, J., Fickas, S., Segall, Z. Disseminating Trust Information in Wearable Communities. In *Proceedings of the 2nd International Symposium on Handheld and Ubiquitous Computing (HUC2K)*, Bristol, UK, 2000.

19. Stepanov, I. Integrating Realistic Mobility Models In Mobile Ad Hoc Network Simulation. Diploma Thesis No. 1989, in English, Department of Computer Science, University if Stuttgart, 2002. Available at ftp://ftp.informatik.uni-stuttgart.de/pub/library/medoc.ustuttgart_fi/DIP-1989/DIP-1989.pdf.

20. Swisscom. Go Mobile. http://www.gomobile.ch/static/en/community/community_web_public.htm

21. Tian., J., Hähner, J., Becker, C., Stepanov, I., and Rothermel, K. Graph based Mobility Model for Mobile ad-hoc Network Simulation. In *Proceedings of the 35th Annual Simulation Symposium (ANNSS35),* San Diego, USA, 2002.

22. Vahdat, A., and Becker, D. Epidemic Routing for Partially Connected Ad Hoc Networks. T*echnical Report CS-200006*, Duke University, USA, 2000.

23. Ward, A., Jones, A., and Hopper, A. New Location Technique for the Active Office. IEEE Personal Communications, 4, 5, 1997, pp. 42-47.

24. Worboys, M. F. *GIS: A Computing Perspective*. Taylor & Francis, London, UK, 1995.

25. Xu, B., Wolfson, O., and Chamberlain, S. Spatially Distributed Databases on Sensors, In *Proceedings of the 8th ACM Symposium on Advances in Geographic Information Systems,* Washington DC, USA, pp. 153-160, 2000.