

Delay-Tolerant Broadcasting

Gunnar Karlsson
Laboratory for Communication
Networks
KTH, Royal Inst. of Tech.
100 44 Stockholm, Sweden
gk@ee.kth.se

Vincent Lenders
Computer Engineering and
Networks Laboratory
ETH Zurich
8092 Zurich, Switzerland
lenders@tik.ee.ethz.ch

Martin May
Computer Engineering and
Networks Laboratory
ETH Zurich
8092 Zurich, Switzerland
may@tik.ee.ethz.ch

ABSTRACT

There are many asynchronous communication situations for which the prevalent continuous connectivity paradigm is not needed. Communication with a fair delay tolerance may instead be provided by intermittent store-and-forwarding between nodes. This paper proposes a design for an open, receiver-driven broadcasting system that relies on delay-tolerant forwarding of data chunks through mobility of wireless nodes. The system provides public broadcast channels, which can be openly used for both transmission and reception. We show by simulation under benchmark mobility models that a delay-tolerant broadcast channel has both a sufficiently high throughput and reach to be interesting as a competitive alternative to the regulated wireless broadcast channel. We also present the design of, and experiences with, a proof-of-concept prototype.

Categories and Subject Descriptors

C.2.1 [Network Architecture and Design]: Distributed networks, Network communications, Store and forward networks, Wireless communication; C.2.2 [Network Protocols]: Applications, Protocol architecture

General Terms

Algorithms, Performance, Design, Experimentation

Keywords

Delay-tolerant communication, wireless broadcast, content distribution

1. INTRODUCTION

Communication networks have traditionally provided continuous end-to-end connectivity. This is natural when the primary service is interactive voice conversations. The continuous connectivity paradigm has however been assumed also for asynchronous services with generous tolerance to delay in the delivery of data. The paradigm might hence be unnecessary, but it comes at a low cost for wired networks. This not the case for wireless networks due

to the intermittencies occurring from insufficient coverage of the areas where the mobile nodes roam; it might be infeasible or uneconomical to provide uninterrupted connections everywhere.

Delay and disruption tolerant communication does not assume that the network consists of more than a set of disjoint point-to-point connections at any given point in time [1]. The data moves from one node to the other, and are spread by the mobility of the nodes. A random network can be seen in hindsight if there would be a record of how messages have been passed among the nodes. Such a loosely connected network might still be of good use for many asynchronous services.

Our generic application is to disseminate content to an arbitrarily large group of receivers, without any assurance on the completeness or order of delivery, in other words, to provide a delay-tolerant broadcasting service. How well this service performs depends on its availability and the movements of the nodes that participate in the content dissemination. As an illustration of the capability of such a system, consider the case where pedestrians are walking in the same direction along a 100-meter long pathway, each with a velocity of one meter per second; the radio transmission supports a constant bit rate of 1 Mb/s within a coverage range of 10 meters forward and backward of the node. If people enter the pathway according to a Poisson process with an average distance of 20 meters between two adjacent persons, then a person makes a pair-wise contact with a person in front or behind with probability 0.56. They share the capacity so the bit rate is 0.5 Mb/s per person. The contact lasts 95 seconds on average and they are able to transfer up to 6 MB during the walk. That is sufficient for more than 6 minutes of music, played back at a rate of 128 kb/s. This simple calculation indicates that it might even be possible to support continuous streaming of audio by delay-tolerant broadcasting since the expected bit rate in this street is 270 kb/s (see our technical report [2] for the detailed derivation of these numbers).

Our main motivation for developing a delay-tolerant broadcasting scheme is to have an alternative to terrestrial and satellite broadcasting systems. We expect the system to be of most use where there is a dense population of users, such as urban areas or gathering spots on, for example, beaches, camping sites, yacht harbors, and in public transportation. We believe that an unlicensed public broadcasting system fills a need since the terrestrial wireless broadcast channels are highly regulated. Not only is the spectrum allocation strictly guarded, but also the concession rights to send programs are severely limited and awarded on politically decided criteria and commercial terms. In addition, the broadcast content may be subject to regulation and sometimes also to censorship. It is therefore interesting to design a wireless broadcast system for the unlicensed ISM bands to support asynchronous and delay-tolerant applications. The system would be open for public transmission

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

SIGCOMM'06 Workshops September 11-15, 2006, Pisa, Italy.
Copyright 2006 ACM 1-59593-417-0/06/0009 ...\$5.00.

from anyone similar to broadcasts on the fixed Internet.

We define broadcast to be a dissemination mode of data for which the group of receivers is completely open; any node that wishes to receive data from a particular broadcast channel is allowed to do so. This is the same definition as that for regular wireless public broadcasts. The distinction between multicast and broadcast might be artificial: we use the term broadcast here only to mean that the group of receivers is undefined and that content spreads indefinitely from node to node. The broadcast is organized in channels (this is analogous to a multicast group), where each channel provides a particular type of content that is selected by a producer. The novelty of this paper is a receiver-driven broadcasting approach, and there is consequently no flooding of content as in epidemic dissemination schemes [3]. A node may entreat content for a channel from a peer node that it meets without the obligation to do so.

The following are the contributions that we report in this paper. We present the concept of the broadcast system and two feasibility studies: a simulation-based study with mobile devices on a plane and a study of the development of a prototype system based on Bluetooth. The paper is structured as follows. The next section describes the concept and the system design in more detail. The evaluation based on simulations is presented in Section 3. A first prototype of a broadcast system for Bluetooth is described in Section 4. Section 5 reviews important related works that have influenced our design, and it discusses our contributions in relation to the prior art. Section 6 summarizes our contributions, presents our conclusion, and outlines further issues of study for making delay-tolerant broadcast a reality.

2. A NEW BROADCASTING SYSTEM

2.1 Concept

The broadcasting system consists of mobile-user nodes in a loosely defined network. Nodes wishing to receive content on a chosen channel solicit it from nodes they encounter as they move around. They might in return be asked for content for the same or any other channel. Nodes also solicit content in order to redistribute it, without the user having a desire to receive it. The solicitation strategy for moving content from node to node allows a node to adapt its caching strategy to its storage and energy resources, or any other local policy; the node may implement various strategies for soliciting content to improve the performance of the system, to solicit content that is requested the most or that the user has most interest in forwarding, for instance.

Taken together, the broadcast and disruption-tolerant paradigms simplify the system: there is no group to be maintained and there is no explicit routing in this communication mode; routing is in fact replaced by mobility as nodes come to places where data are available. All data transfers are one-hop long and the link layer thus resolves contention for the data transmission between nodes. Our systems architecture leverages all types of wireless data-link protocols, for instance IEEE 802.11 and Bluetooth.

Contents are brought into an area of interest by the mobile nodes, or, alternatively, they are generated by the mobile nodes in the area. The mobile nodes could retrieve content when they "dock" at places where they are connected to the Internet (for instance, by means of podcasting). The content retrieval could also be over a wireless network, such as a wireless local-area network or a wide-area cellular network. Our broadcasting system adds the following advantages to podcasting and to multicasting over wireless networks.

1. It provides contents to mobile nodes between the docking

opportunities (which could be hours apart).

2. It extends the reach of the dissemination of data in the wireless network.
3. Contents that originate from the mobile nodes could be broadcast without any infrastructure.

Hence, the delay-tolerant broadcast complements infrastructure-based broadcasts, and offers a new mode of ad hoc broadcasting between mobile nodes for any content that they generate and which should be openly disseminated (such as for example still images, video, and voice clips). The connection between the delay-tolerant broadcast and the infrastructure is not in the focus of this work, and we do not further discuss how content is initially brought into the wireless domain in this paper.

The data is delivered in "chunks" which could be delivered in or out of order, with or without assured completeness. Order can be imposed at the receiving end either by soliciting chunks in a given order or, better, by soliciting them in any order, followed by sorting. We will therefore be concerned with the possible reception rate for a mobile node, as well as the dissemination rate and reach of an individual chunk of data. Note that there are many applications for which order and completeness are not necessary: the distribution of a mixture or music, news, traffic and weather information, as that broadcast by most radio stations today, would not require complete nor orderly delivery, as long as the data chunks correspond to meaningful parts of content, a song or a news item for instance.

A mobile node that has associated with a peer node may solicit content for the particular broadcast channels that it listens to, or for any channel for which it is willing to carry content. Since there is no connected network in the wireless domain, there is no need for routing. The application is hence based either on UDP over IP for which the packets are broadcast with TTL set to one (the broadcast could be for a specific sub-network, if appropriate), or on the data-link protocol directly (as in our prototype). The UDP/IP suite might be useful for docking, but places where the node remains resident for longer times could equally well provide systems running the broadcasting protocol so that the docking is at the application layer.

We assume that nodes connect pair-wise and that neighboring pairs do not disturb each others, e.g., they use different channels for 802.11 or different sequences for the frequency hopping for Bluetooth. The data link association is established according to the specification of the particular link layer being used, and the MAC protocol resolves contention when nodes meet; a node does not associate with more than one other node. The reason is to economize on the limited connections to maximize the amount of data transferred (it is also supported by the optimal transmission range [4]). The system could use robust header compression, provided that it actually speeds up the transfer when the association times are short. We do not address problems at the physical layer, such as interference, decay or fading. If the radio conditions are good, a connection is established, otherwise it is not; interference is dealt with through the proper design of the link layers. The design issues for the broadcast lie instead at the application layer: the solicitation protocol, the naming schemes for content and broadcast channels.

2.2 System design

In the following, we describe our system design. We first describe how broadcast channels are structured. Then, we introduce how nodes discover and request content from channels. We also discuss how our delay-tolerant broadcasting system could be improved by deploying fixed caching nodes at strategic locations.

2.2.1 Channels and chunks

A broadcast channel is defined by a name, some metadata about its content, and the name of the producer, followed by a list of chunk names including their sequence numbers. All broadcast channels and data chunks have unique universal resource identifiers (URI) [5]. Given a channel name, it is possible to locate the definition file of the channel. A chunk is defined by its name, its file type, and some metadata. The metadata may for example include artist and title to be displayed at playback, and can also be used for searches. An important issue is to determine what constitutes official content for a channel: for instance music of a certain genre, or from a selected group of artists. This is the decision of the producer for the channel. However, our system also allows for channels where any content of a specific media type is allowed: for instance a music channel for which everybody may provide content.

2.2.2 Channel announcements

In order to bootstrap the listening, one channel is pre-defined: the *discovery channel*. This channel announces the names of broadcast channels and it is carried by all nodes. A node that does not know what channels to ask for makes a request for this particular channel and, as reply, would get a chunk containing the list of channels known by the corresponding node. The user then selects a set of channels from this list for which it asks for content of interest from nodes that it will meet.

2.2.3 Request for content

The protocol for requesting content is straightforward: a node simply requests content from a set of channels, or it requests any of a set of specified chunks of data. The following should apply:

- A named chunk may be requested with a given offset to allow the completion of a previously interrupted transfer.
- Request for content from a channel may identify chunks that are not of interests (for instance those that already are retrieved).
- If node A solicits content for channel i and node B only stores content for channel j , then B should provide A with any chunk that appears on both channels i and j .
- A chunk of data may be requested based on its metadata.

A request is issued as a list: $\langle name1 \rangle, \langle offset \rangle ; \langle name2 \rangle, \langle offset \rangle ; \langle nameN \rangle, \langle offset \rangle$. Each name is a URI for either a chunk, in which case the offset is measured in bytes within the data volume, or for a channel and the offset is then the lowest sequence number of interest for a chunk (or the lowest time stamp) on the channel. Requests for channels and chunks might be mixed in a request and the list may be of arbitrary length; it should ideally fit in a single packet. Content may also be requested based on meta-data. The third bullet point is difficult to ensure: it would require chunks to carry all the channel names where they appear, but the channel content is selected by producers who are not coordinated with one another. We will hence not solve this issue at this point.

The associations when nodes meet might be short and it is vital to economize on the connection time. The most efficient implementation is to embed the requests for data in the beacons. However, it would require a cross-layer design that might be difficult to have accepted for MAC layer standards; it would for instance cause more overhead since the beacon frames would be longer than otherwise.

2.2.4 Fixed caching nodes

The performance of the broadcasting system depends on the density, on the mobility, but also on the availability of content at the nodes. To improve the performance in general, we propose the use of dedicated fixed nodes acting as caches. These caching nodes would typically not have an application for the data they store and would only serve to disseminate the content when the density of mobile nodes is low. Since fixed caching nodes do not have an Internet connection, they may easily be placed in public transportation systems and throughout the built environment. Note however, that the fixed caching nodes are optional in our design, and only introduced to improve the overall performance of the broadcasting.

3. SIMULATION STUDY

In this section, we analyze a delay-tolerant broadcasting system in a two-dimensional topology where people are moving on a square. The goal of the study is to see if the broadcast system appears feasible for reasonable system parameters. The remainder of this section is structured as follows. We first describe our simulation model and the assumptions. Then, we look at the distribution of the association times and inter-association times. The performance of the system is finally evaluated, using the average per node bit rates and the content dissemination rates as metrics.

3.1 Simulation model

We used our own simulator with a very simple communication model: nodes communicate with each other at a nominal bit-rate b if their geometric distance is smaller than a threshold value (the wireless range of the radio devices). More sophisticated simulation tools, such as Glomosim or ns-2 which incorporate lower layer protocol details, are not necessary for our study; we do not aim at analyzing protocol performance details, but rather aim at identifying the feasibility of delay-tolerant broadcasting.

We assume an environment where humans carry Bluetooth enabled devices and are moving on a square. The mobile nodes are moving with a speed v of $1m/s$ (this represents approximately the average speed of humans), the radio range r of each device is 10 meters (the wireless range of class 3 Bluetooth devices), and the wireless channel bit rate b is of $1Mb/s$. The square size is set to 300 by 300 meters and the number of nodes in the square is fixed during the whole simulation time. In our simulation setup, the nodes are not in range of each other most of the time. Nodes only occasionally come into wireless proximity and associate. Each node associates with only one neighbor at a time. Therefore, an already associated node declines any additional association request from another node.

The time required to setup an association is referred to here as the setup time T . This time includes the time (i) to awake the nodes from sleeping mode; (ii) to synchronize the two devices; and (iii) to setup a connection. No data is transferred during the setup. Two nodes stay associated with each other until all desired data has been exchanged, or until the nodes move apart so that they are outside the communication range. We simulate user mobility with two benchmark models: the random waypoint (RWP) [6] and the random walk (RW) [7] mobility models.

Note that for the metrics we are interested in, a group mobility model delivers the same results as for a single node mobility model since the group has synchronized its content after some time and behaves as one larger node with a slightly increased transmission range. Afterwards, the group mobility model has similar properties as a single node mobility model such as the RWP or the RW model and hence, groups of nodes do not contribute to faster content spreading. Note further that we initially place the nodes with

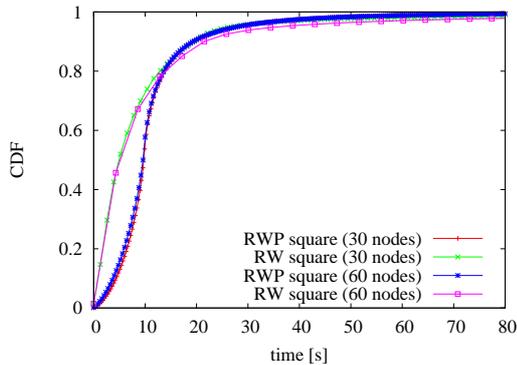


Figure 1: CDF of the association duration.

the RWP model on the square according to [8] to make sure that the node distribution is in steady state over the whole simulation time.

3.2 Association durations and rates

First, we analyze the durations and the rates of associations for both mobility models. These fundamental values help understanding the average bit rates and the data dissemination rates we discuss later in this section.

We assume that the nodes stay associated with each other until they move apart (this is the case when associated nodes have content to share during the whole association time). We consider two different node densities (30 and 60 nodes) to examine the effect of the population density in the square area (these densities result in a maximum wireless coverage of approximately 10% and 20%, respectively, of the area).

The cumulative distribution functions (CDF) for the durations is plotted in Figure 1. The duration of an association is defined as the time interval between the moment two un-associated nodes come into wireless range until the moment they leave. We observe that the mobility models produce quite different distributions of the contact durations.

In the RWP model, the most probable duration is of about 10 seconds. This corresponds to the association time of two people moving in opposite directions ($\frac{2r}{v} = 10[s]$). Larger time values arise when two nodes move in the same direction for a certain amount of time. In contrast, the RW mobility model produces most contact durations in the range between zero and ten seconds. Regarding node density, we conclude that the node density does not have a major impact on the duration of associations for either benchmark mobility models.

We next look at the rate of associations. For this, we plot in Figure 2 the CDF for the time interval from the moment a node loses an association until it re-associates with another (or the same) node. We observe that the time period is on average shorter for the RW model. Thus, we conclude that associations with the RW model are shorter than with the RWP model, but occur more frequently. The open question, we address later in this section, is if this property leads to higher average bit rates compared to those obtained with the RWP model.

An additional interesting issue becomes visible when comparing the inter-association rates with different node densities (30 vs. 60 nodes). One would expect that the rate of intermittent associations increases when the node density increases. This is indeed the case for the RWP model. However, for the RW model, the difference is only marginal and hardly visible in the plot. This is due to the more localized mobility pattern of the RW model. Nodes tend to move

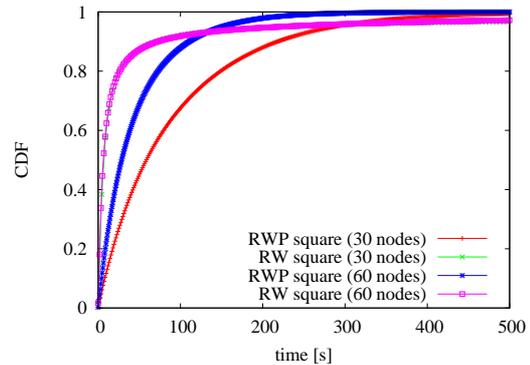


Figure 2: CDF of the inter-association time.

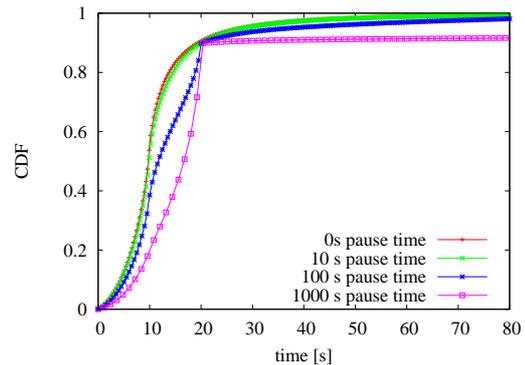


Figure 3: CDF of the association duration for different pause times with the RWP model.

rather locally compared to the RWP model.

It is unrealistic to assume that people are constantly moving without pausing from time to time. Therefore, an important aspect we investigate is how the node pause times impact the association rates and durations. We plot the results of our simulations for the RWP model with various pause times in Figures 3 and 4. We assume that each node waits for a fixed pause time when it arrives at a way point. We see that the pause time significantly affects the duration and rates of associations. Pausing nodes produce longer contact durations but less frequent associations. Consider for example the curve with a pause time of 1000 seconds. Many associations last for around 20 seconds compared to 10 seconds for 0s pause time. 20 seconds occur when one device is pausing and another device passes straight across its transmission region. We also see relatively many associations which are larger than 20 seconds for pause times of 1000s when comparing to simulations with smaller pause times. These occur mainly when two nodes pause at the same time while being within wireless range. Again, an open question is how the pausing of nodes impacts the average bit rate and the dissemination rate of content. We answer this in the next subsections.

3.3 Average bit rates

So far, we examined how the mobility model, the node density, and the pause time affect the association durations and inter-association times. Now, we discuss how these parameters affect the average transmission bit rates between the nodes. The specific metric we look at, is the maximally achievable average bit rate per node for content downloads. For this analysis, we assume the following

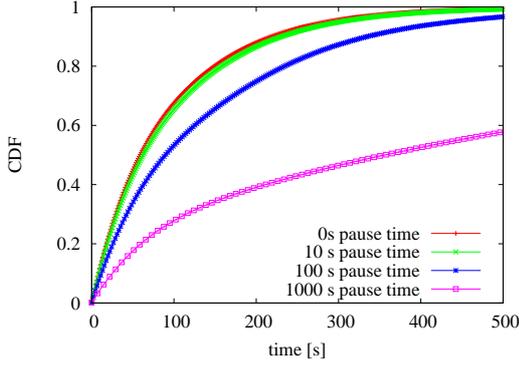


Figure 4: CDF of the inter-association time for different pause times with the RWP model.

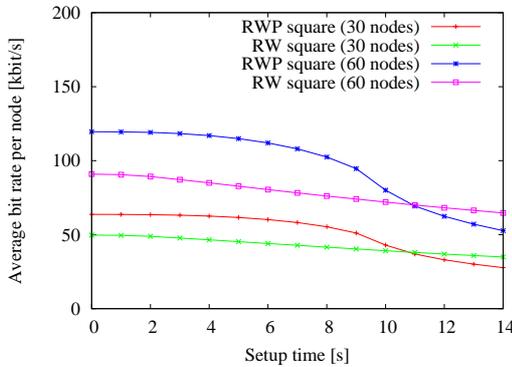


Figure 5: Average bit rate for both mobility models.

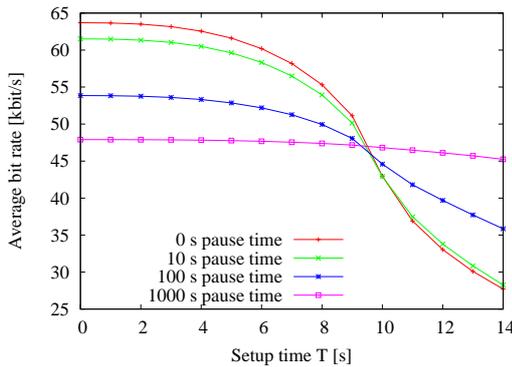


Figure 6: Average bit rate against setup times for different pause times, RWP mobility, and 30 nodes.

scenario:

- Two nodes share the wireless capacity of an association in a fair manner. The maximal capacity per node is 0.5 Mb/s when downloading simultaneously.
- In all associations, both peers download content at full data rates (0.5 Mb/s in both directions).
- There is no interference between associations. Since we only consider sparsely connected networks, the assumption that associations are not considerably interfering with each other is sound.

The bit rates, averaged over all nodes, for both mobility models are plotted in Figure 5. The bit rates are plotted against the setup time on the x-axis. Recall that the setup time T is the time interval between the moment when two nodes come into wireless range and the moment the association is initiated to transmit data. Note that in our Bluetooth-based prototype system (see Section 4), we frequently measured setup times mostly up to 10 seconds, but sometimes even larger.

From the figure, we draw three main conclusions: First, even in a sparsely populated network (30 nodes in the 300 by 300 meters square), the nodes achieve reasonable average bit rates around 50 kb/s for both mobility models. Second, the performance with both mobility models decreases slightly for setup times between 0 and 8 seconds. For setup times larger than 10 seconds, the bit rates with the RWP drop down sharply, even below the bit rates obtained with the RW mobility model. Last, the node density has a large influence on the per node bit rates. By doubling the density, the average per node bit rates increase by almost 90% for both mobility models.

We also examined how the pausing of nodes affects the average bit rates. For this, we determined the bit rates for pause times of 0, 10, 100 and 1000 seconds with the RWP model for 30 nodes. The results are plotted in Figure 6. For small setup times, the bit rates are largest when the nodes are constantly moving. This is because nodes even profit from short associations which occur frequently with high mobility. However, as the setup time increases, the achieved bit rates decrease to a similar value for all four pause times (between 9s and 10s). For very large setup times, the longer the pause time, the better is the achieved average bit rate.

The reason is that it is no longer possible to profit from small associations which occur when the nodes are moving in opposite directions. Data is transferred mainly between node pairs where one of the two nodes is pausing. The nodes then profit from the increased contact duration. We also see that, in contrary to what one would expect, large pause times do not degrade the bit rates significantly even for very small setup times. For an ideal setup time of 0 seconds, the bit rates with 1000 seconds pause time are only 25% smaller than those obtained with 0 second pause time.

3.4 Content dissemination rate

We now discuss the other of the key metrics of delay-tolerant broadcast systems: the content dissemination rate. This metric captures the time required until a fraction of the nodes receive a new piece of content over the broadcast channel. In sparsely populated areas, as those used in our studies, the dissemination rate is determined mainly by the node speed, the node movement (determines the area covered and the order of the node encounters), as well as the per node average bit rates we studied before.

In this simulation scenario, we assume data chunks with a size of 20 kB. A data chunk is the smallest data unit. If an association breaks before a complete chunk is received, the whole chunk must be re-sent in a following association (note that a 20 kB transfer only

takes 320 ms at a bit rate of 0.5 Mb/s as used in the simulations). We determine the time necessary to broadcast 200 kB (10 chunks) and 2 MB (100 chunks) respectively. While the probability that 200 kB are sent over a single association is high with both mobility models, the probability that 2 MB are sent over a single association is very low. Therefore, the 2 MB broadcasts setup aims at studying the dissemination rates involving multiple consecutive associations. We start by analyzing the dissemination rate in an ideal case where the setup time T is zero seconds, and the nodes do not pause. The results in seconds with a 95-percent confidence interval are given in Table 1.

We distinguish between the average time to broadcast the content to 10%, 50%, 90% and 100% of the nodes with the RWP and RW models. We identify three main trends: (i) as expected, larger node densities help disseminating the content faster; (ii) due to the rather local mobility patterns of the RW model, the time to broadcast is much larger than with the RWP model. Indeed, transmission delays are in the order of one magnitude higher than with the RWP model, even for large data quantities (2 MB) to a small fraction of nodes (10%). This result is surprising since the average bit rates are in the same order of magnitude for both mobility models. And (iii) as foreseen, the small amounts of contents are disseminated much faster than large ones.

In Table 2, we analyze the impact of the pause time with the RWP model on the dissemination time. The table entries are in seconds with a 95-percent confidence interval. Here, the main finding is that large transfers profit most with longer pause times. The impact of the setup time on the dissemination performance is illustrated in Table 3 (again in seconds with a 95-percent confidence interval). As expected, the shorter the setup time, the better the dissemination rate. Surprisingly, the dissemination rate with longer setup times ($T = 10$ s) is better with long pause times. Recall that long pause times result in longer average contact durations and hence, the longer setup time can thus be compensated.

3.5 Conclusion and discussion

We simulated the performance of the delay-tolerant broadcasting scheme for sparsely populated squares. We have shown that for a decent node speed (v of $1m/s$), the per node download bit rates with both mobility models are fairly high (between 50 and 120 kb/s). Even when the setup times of associations become so large that small contacts of two nodes moving in the opposite direction can no longer be used, the average bit rates do not decrease too extensively.

Regarding content spreading, we have seen that for rapid dissemination among the nodes, it is necessary that some nodes move over large distances frequently as it is the case with the RWP model. Then, it is for example possible to disseminate 200kB to 90% of the nodes within less than 7 minutes. Note that in the optimal case, content is spread with the speed of the mobile nodes, hence 5 minutes from one side of the square to the opposite side. For a more local mobility pattern as the RW model, much larger node densities are required to achieve similar dissemination rates.

Pausing nodes reduce the amount of small contacts and increase the amount of large contacts. In total, the effect is minimal for both the average bit rates and the dissemination rates. For large setup times, the performance is slightly better when the nodes are pausing, whereas for small setup times, the performance is slightly better when the nodes are constantly moving. (Fixed caching nodes may be seen as mobile nodes with indefinite pause times and their usefulness are indicated by the results for the 1000s pause time.)

4. PROOF-OF-CONCEPT PROTOTYPE

In order to assess the feasibility of a delay tolerant broadcasting system, we developed a prototype application for mobile phones called *Bluetella*.

4.1 Basic concepts

Bluetella is an application that runs on mobile devices. To exchange data, two Bluetella devices establish a point-to-point connection via wireless Bluetooth communication. To participate in a Bluetella network, a node simply has to run the Bluetella application and announce a Bluetella specific profile UUID. Bluetella communication is divided in three phases: environment scanning, state synchronization, and data transfer. First, simultaneously with the profile announcement, the device starts scanning the environment for other Bluetella enabled devices.

Whenever two Bluetella nodes are in transmission range, they establish a connection and start synchronizing. In this second step, both nodes send the list of channels or content they are interested in to the neighbor node. If content of the requested channel is available, the node starts transmitting data back to the requesting device (phase three). Even if the requested content is not available, the nodes keep track of the requested channels/content, maintaining a global hit list of the most popular channels/content. During the synchronization phase, after the exchange of the search lists, the nodes also exchange these global hit lists and re-compute, with their own list, an updated global hit list. After the regular content is downloaded, devices with larger storage capacities then may download additional content from the global hit list if available. This function is of particular interest for fixed nodes serving as content caches as described in Section 2.

4.2 Experiences with the Bluetella system

The prototype application is implemented on the Java 2 Micro Edition Platform (J2ME). To run the application on the cell phones, they have to support MIDP 1.0, as well as the Bluetooth API specified in JSR-82 [9]. We tested Bluetella on multiple cell phones, such as the Nokia 6630 and the cell phones of the Siemens S-Series. Based on the specific phone hardware, the size of the application was about 70 kB and the application allocated between 700 kB and 1500 kB memory, depending on the number of connections and the size of the request and hit lists. The major problem we encountered with our DTB application was the lengthy discovery time of Bluetooth. According to the Bluetooth specification, "the inquiry substate may have to last for 10.24 seconds unless the inquirer collects enough responses and determines to abort the inquiry substate earlier" [10].

If multiple devices are scanning the environment for devices, the scanning time might even increase to higher values. In theory, Bluetooth provides bandwidth up to 721 kb/s on an asymmetric channel. In practice, the throughput is considerably less once protocol overhead and error control are factored in. In the measurements we conducted, we achieved download speeds between 39.2 kb/s and 70 kb/s. There are three lessons we learned from this prototype implementation: (i) delay tolerant broadcasting systems are easy to develop and deploy; (ii) the main concern with Bluetooth communication for DTB system is the long communication setup time; and (iii) the communication rates are high enough for small data transfers; for larger files, data/header compression is beneficial.

5. RELATED WORK

The area of delay-tolerant networks is gaining attention and the research is progressing in many different directions. The Delay Tolerant Network Research Group (DTNRG) [11] has proposed an ar-

	10%		50%		90%		100%	
	200k	2M	200k	2M	200k	2M	200k	2M
RWP 30 nodes	193 ± 10	1283 ± 31	400 ± 11	1583 ± 28	593 ± 12	1801 ± 28	783 ± 16	2011 ± 33
RWP 60 nodes	128 ± 6	848 ± 26	258 ± 7	996 ± 25	388 ± 8	1129 ± 26	563 ± 10	1300 ± 28
RW 30 nodes	4810 ± 1148	9958 ± 1920	13850 ± 1176	22531 ± 2382	23848 ± 2156	35024 ± 2789	34317 ± 2819	44061 ± 3490
RW 60 nodes	2011 ± 146	3598 ± 480	5766 ± 224	8441 ± 533	9484 ± 269	13061 ± 671	13240 ± 369	17577 ± 846
RW 150 nodes	751 ± 119	1306 ± 51	2199 ± 228	3177 ± 75	3450 ± 277	4840 ± 83	4305 ± 328	6029 ± 99

Table 1: Time required in seconds to broadcast content (200kB and 2MB) to 10%, 50%, 90%, and 100% of all nodes.

	10%		50%		90%		100%	
	200k	2M	200k	2M	200k	2M	200k	2M
0s pause time	193 ± 10	1283 ± 31	400 ± 11	1583 ± 28	593 ± 12	1801 ± 28	783 ± 16	2011 ± 33
10s pause time	190 ± 11	1331 ± 36	409 ± 12	1642 ± 33	611 ± 13	1870 ± 33	810 ± 17	2090 ± 36
100s pause time	313 ± 19	1565 ± 61	631 ± 22	2031 ± 56	918 ± 25	2373 ± 56	1218 ± 32	2701 ± 61
1000s pause time	1461 ± 110	5560 ± 441	2755 ± 134	7459 ± 409	3988 ± 150	8671 ± 484	5216 ± 187	10133 ± 572

Table 2: Time required in seconds to broadcast content (200kB and 2MB) with the RWP model to 10%, 50%, 90%, and 100% of all nodes.

	T=0s		T=5s		T=10s	
	200k	2M	200k	2M	200k	2M
0s pause time	593 ± 12	1801 ± 28	770 ± 23	2765 ± 146	2015 ± 40	6496 ± 150
10s pause time	611 ± 13	1870 ± 33	788 ± 19	2897 ± 88	1965 ± 38	6367 ± 144
100s pause time	918 ± 25	2373 ± 56	1094 ± 35	3252 ± 108	1843 ± 39	5635 ± 163
1000s pause time	3988 ± 150	8671 ± 484	4245 ± 187	10835 ± 1222	5521 ± 113	16439 ± 571

Table 3: Time required in seconds to broadcast content (200kB and 2MB) to 90% of all nodes for different pause times and different setup times T.

chitecture [12] to support messaging that may be used by delay tolerant applications. The architecture consists mainly of the addition of an overlay, called the bundle layer, above a network transport layer. Messages are transferred in bundles in an atomic fashion between nodes, using a transport protocol that ensures node-to-node reliability. These messages can be of any size. Nodes are assumed to have buffers in which they store the bundles. In contrast to this approach, we do not aim at providing reliable data transfer.

Routing in delay-tolerant networks has been addressed recently in [3], [13], [14],[15], [16]. Our system does not require a routing infrastructure. Our system is a pure interest-based pull approach where nodes only download content from one-hop neighbors. Multicast routing in DTN has been addressed in [17]. While the goal of our system is also to deliver data to a group of people, our approach is decoupled from any multicast semantics (such as group memberships, et cetera). The work by Choudhury [18] also concerns data dissemination, albeit it assumes a random push mode (gossip).

Data collection in partially connected sensor networks has been addressed in [19] [20], [21]. The basic idea of these works is similar to ours, namely exploiting mobility to forward data in the network. The main difference is that sensor networks aim at pushing information out of the network to data sinks, whereas our system is designed to pull content down to the mobile user. Khelil et al. [22] suggest several metrics for quantifying the impact of mobility at large time-scale on the performance of delay-tolerant networks. We have used some of those metrics in our work. Hui et al. have conducted measurements with real human mobility in [23]. Lenders et al. [24, 25] measured human mobility in an office environment. Such traces could be used to validate our system for real situations.

The "drive-through internet" is based on WLAN coverage along roads to provide intermittent network access; there is no relaying of data between cars [26]. The concept of mobile infostation networks is equivalent to delay-tolerant networks; the work presented

in [27] lends support to our design. Also, it has been shown that the long-term throughput for an ad hoc network with mobile nodes can remain constant as the density of nodes increases [28]. And finally, Peoplenet has some similarities with our approach in the way information propagates on a contact basis between mobile users [29]. However, Peoplenet heavily relies on a fixed infrastructure and is targeted at seeking information in contrast to broadcasting information.

6. CONCLUSION

We have proposed the concept of delay-tolerant broadcasting to allow wide and public dissemination of data over radio. It enhances broadcasting based on an infrastructure, and offers a new ad hoc distribution mode for contents that originate from mobile nodes. Our approach is purely receiver driven. There is no flooding or explicit routing in the network. Content for broadcast channels is spread through node mobility and peer-to-peer contact between pairs of neighbor nodes. The system for achieving this is not complex; as our prototype implementation shows, the program is small and runs on mobile phones. The feasibility is rather connected to the performance of the system. We have studied it for mobility in two dimensions through simulation. We find the results encouraging and expect the concept to be workable in urban areas and other places where the node density is reasonably high. We will hence continue with the design of the system and evaluate it further by also considering traces from measurements [23, 24, 25]. The study will also address the issue of authentication so that a node can verify that a chunk it received indeed belongs to the channel for which it requested content.

ACKNOWLEDGMENTS

Gunnar Karlsson thanks Prof. Bernhard Plattner for the possibility of spending a year's sabbatical at ETH Zurich, where this work was

conducted. The authors thank Matthias Bossardt, Andreas Weibel, and Lukas Winterhalder of ETH Zurich for their contribution regarding the implementation of the Bluetella prototype.

7. REFERENCES

- [1] Kevin Fall, "A Delay-Tolerant Network Architecture for Challenged Internets," in *Proceedings of the ACM SIGCOMM*, Karlsruhe, Germany, August 2003.
- [2] Gunnar Karlsson, Vincent Lenders, and Martin May, "Delay-Tolerant Broadcasting," Tech. Rep. TRITA-EE 2006:021, KTH, Royal Institute of Technology, May 2006.
- [3] A. Vahdat and D. Becker, "Epidemic Routing for Partially Connected Ad Hoc Networks," Technical Report CS-200006, Duke University, NC, USA, April 2000.
- [4] W. H. Yuen and R. D. Yates, "Optimum transmit range and capacity of mobile infostation networks," in *Proceedings of IEEE GLOBECOM*, San Francisco, USA, December 2003.
- [5] T. Berners-Lee, R. Fielding, and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax," RFC 3986, January 2005.
- [6] David B. Johnson, David A. Maltz, Yih-Chun Hu, and Jorjeta G. Jetcheva, "The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks (DSR)," February 2002.
- [7] Tracy Camp, Jeff Boleng, and Vanessa Davies, "A Survey of Mobility Models for Ad Hoc Network Research," in *Wireless Communications and Mobile Computing (WCMC): Special issue on mobile ad hoc networking: research, trends and applications*, 2002, vol. 2, pp. 483–502.
- [8] W. Navidi and T. Camp, "Stationary distributions for the random waypoint mobility model," *Transactions on Mobile Communication*, vol. 3, pp. 99–108, 2004.
- [9] Sun Microsystems, "Java apis for bluetooth wireless technology - jsr-82," <http://www.jcp.org/en/jsr>.
- [10] The Bluetooth Special Interest Group, "Specification of the Bluetooth system: Part B Baseband Specification," December 1999.
- [11] "Delay Tolerant Network Research Group (DTNRG)," <http://www.dtnrg.org>.
- [12] S. Burleigh, A Hooke, L. Torgerson, K. Fall, V. Cerf, B. Durst, and K. Scott, "Delay-tolerant Networking: An Approach to Interplanetary Internet," *IEEE Communications Magazine*, vol. 41, no. 6, pp. 128–136, 2003.
- [13] F. Bai and A. Helmy, "Impact of mobility on mobility assisted information diffusion (maid) protocols," *Technical Report*, vol. Dep. Electrical Eng., University of Southern California, 2005.
- [14] Matthias Grossglauser and Martin Vetterli, "Locating Nodes with EASE: Last Encounter Routing in Ad Hoc Networks through Mobility Diffusion," in *Proceedings of IEEE INFOCOM*, San Francisco, USA, March 2003.
- [15] Sushant Jain, Kevin Fall, and Rabin Patra, "Routing in a Delay Tolerant Network," in *Proceedings of the ACM SIGCOMM*, Portland, Oregon, USA, August 2004, pp. 145–157.
- [16] S. Jain, M. Demmer, R. Patra, and K. Fall, "Using redundancy to cope with failures in a delay tolerant network," *Proceedings of ACM SIGCOMM*, 2005.
- [17] W. Zhao, M. Ammar, and E. Zegura, "Multicasting in delay tolerant networks: Semantic models and routing algorithms," in *Proceedings of the Sigcomm Workshop on Delay Tolerant Networking*, August 2005.
- [18] R. R. Choudhury, "Brownian gossip: Exploiting node mobility for diffusing information in wireless networks," in *Proceedings of the Workshop on Stochasticity in Distributed Systems*, 2005.
- [19] Philo Juang, Hidekazu Oki, Yong Wang, Margaret Martonosi, Li-Shiuan Peh, and Daniel Rubenstein, "Energy-Efficient Computing for Wildlife Tracking: Design Tradeoffs and Early Experiences with ZebraNet," in *Proceedings of the Tenth International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS-X)*, San Jose, CA, USA, October 2002.
- [20] R. C. Shah, S. Roy, S. Jain, and W. Brunette, "Data mules: Modeling a three-tier architecture for sparse sensor networks," in *Proceedings of IEEE SNPA Workshop*, 2003.
- [21] W. Zhao, M. Ammar, and E. Zegura, "A message ferrying approach for data delivery in sparse mobile ad hoc networks," in *Proceedings of ACM Mobihoc*, 2004.
- [22] A. Khelil, P. J. Marrón, and K. Rothermel, "Contact-based mobility metric for delay-tolerant ad-hoc networking," in *Proceedings of IEEE MASCOTS*, 2005.
- [23] Pan Hui, Augustin Chaintreau, James Scott, Richard Gass, Jon Crowcroft, and Christophe Diot, "Pocket Switched Networks and Human Mobility in Conference Environments," in *Proceedings of the ACM SIGCOMM 2005 Workshop on Delay-Tolerant Networking (WDTN)*, Philadelphia, Pennsylvania, USA, August 2005.
- [24] Vincent Lenders, Jörg Wagner, and Martin May, "Analyzing the Impact of Mobility in Ad Hoc Networks," in *ACM/Sigmobile Workshop on Multi-hop Ad Hoc Networks: from Theory to Reality (REALMAN)*, Florence, Italy, May 2006.
- [25] Vincent Lenders, Jörg Wagner, and Martin May, "Measurements from an 802.11b Mobile Ad Hoc Network," in *Proceedings of the IEEE WoWMoM workshop on advanced experimental activities on wireless networks and systems (EXPONWIRELESS)*, Niagara-Falls/Buffalo, New York, USA, June 2006.
- [26] J. Ott and D. Kutscher, "A Disconnection-Tolerant Transport for Drive-thru Internet Environments," in *Proceedings of IEEE INFOCOM*, Miami, USA, March 2005.
- [27] W. H. Yuen, R. D. Yates, and S. C. Mau, "Exploiting data diversity and multiuser diversity in noncooperative mobile infostation networks," in *Proceedings of IEEE INFOCOM*, San Francisco, USA, March 2003.
- [28] M. Grossglauser and D. Tse, "Mobility increases the capacity of ad hoc wireless networks," in *ACM Transactions on Networking*, 2002.
- [29] M. Motani, V. Srinivasan, and P. Nuggehalli, "Peoplenet: Engineering a wireless virtual social network," in *Proceedings of ACM MobiCom*, 2005.