Smart-tag Based Data Dissemination

Bonnet, Philippe; Beaufour, Allan; Leopold, Martin

Published in:
Proceedings of the First ACM International Workshop on Wireless Sensor Networks and Applications

DOI:
http://doi.acm.org/10.1145/570738.570749

Publication date:
2002

Document version
Publisher's PDF, also known as Version of record

Smart-Tag Based Data Dissemination

Allan Beaufour, Martin Leopold, Philippe Bonnet
Institute of Computer Science
University of Copenhagen
beaufour@diku.dk, leopold@diku.dk, bonnet@diku.dk

ABSTRACT

Monitoring wide, hostile areas requires disseminating data between fixed, disconnected clusters of sensor nodes. It is not always possible to install long-range radios in order to cover the whole area. We propose to leverage the movement of mobile individuals, equipped with smart-tags, to disseminate data across disconnected static nodes spread across a wide area. Static nodes and mobile smart-tags exchange data when they are in the vicinity of each other; smart-tags disseminate data as they move around. In this paper, we propose an algorithm for update propagation and a model for smart-tag based data dissemination. We use simulation to study the characteristics of the model we propose. Finally, we present an implementation based on Bluetooth smart-tags.

1. INTRODUCTION

Data dissemination can be defined as the proactive distribution of relevant data to large numbers of users [4]. The objective is to deliver *the right data, to the right people at the right time*. In the context of sensor networks, data dissemination is a natural basis for monitoring applications.

The mechanisms that underly data dissemination are, first, the matching of data to user interests, and second, the delivery mechanisms that ensure distribution of data to users. In this paper, we focus on the latter. We study a delivery mechanism relying on mobile smart-tags to distribute data to fixed user nodes spread across a wide area. We assume a trivial user request matching mechanism whose goal is that data should be distributed to all user nodes. In other words, sensor data should be replicated to all users. Incorporating complex user requests with the data delivery mechanism we describe in this paper is a topic for future research.

It is generally assumed that wireless sensor networks rely on dense deployments of sensor nodes. A dense deployment is the condition for the establishment of a multi-hop network, where data is routed through multiple nodes on the way from sources to sinks, thus using short-range radio to save energy [21]. The main challenge with such wireless networks is to account for the transmission cost when performing various forms of in-network processing such as collaborative signal processing [27] or data aggregation [8]. In some scenarios, clusters of sensor nodes are connected via a backbone (say the Internet) [6].

In this paper, we study an alternative form of system where a few sensor nodes are scattered over a wide area [1]. Let us consider for instance a national park where weather stations are located at camping places all over the park; electronic display boards are placed along the roads taken by the hikers. The park is big enough, so that it is not a viable solution to cover it completely with wireless Ethernet or to equip each weather station or each display board with long range radio. An interesting solution consists in (a) equipping sensor nodes and display units with short-range radios and (b) equipping hikers with radio enabled smart tags, so that they can disseminate data as they move around. As a hiker walks by a weather station, his smart tag collects the latest measurements; he later transmits them to the display boards along the path he is following. A second hiker crossing this path gets her smart-tag updated and further disseminates data along her path. Note that in this scenario, data dissemination is achieved via a sequence of point-to-point updates between fixed and mobile devices (as opposed to multihop communication in a dense sensor network).

We consider systems where a large number of mobile smart-tags disseminate data from fixed sensor nodes to fixed display units as they move around. We believe that this form of data dissemination is relevant when monitoring large, hostile areas populated by individuals (humans, animals or robots) that can be equipped with smart-tags. For instance, environmental research is a natural target, with sensor nodes scattered over hundreds of square miles, and animals equipped with smart-tags in order to disseminate information across the area. Military applications are another candidate with possibly teleguided flying devices equipped with smart-tags connecting sensor nodes scattered over a large area [3]. In our example, hikers spread data across a national park as they move around. This example raises the following questions:

- *Coverage*: is data disseminated from all sensor nodes to all display units?
- *Delay*: how fast is data transmitted from a sensor node to a display unit?

In order to answer these questions, an approach is to consider smart-tag based data dissemination as a form

[1] A source is a sensor node that generates measurements or detections, a sink is another sensor node or possibly a gateway to a wide-area network infrastructure.

WSNA'02, September 28, 2002, Atlanta, Georgia, USA.

of epidemics. Data sources are similar to *infectious* individuals and display units are similar to *susceptible* individuals. The contacts between individuals occur indirectly through the movement of smart-tag between static nodes.
The form of data dissemination we consider is however not modeled by simple models such as SIR (Susceptible, Infectious, Removed [2]). In a SIR model, a single disease is propagated in a population where individuals are at first susceptible to the disease, then possibly infected, i.e. infectious for some time (they might infect the individuals they make contact with) and later on immune. First, such SIR models do not allow to model multiple diseases (data sources); second, in our model nodes remain infectious indefinitely; third, SIR models essentially focus on the spread of the epidemic while we are also interested in the time it takes for a node to be updated.
In this paper, we have chosen not to develop a complete stochastic model for smart-tag based data dissemination. This is a topic for future research. Instead we use simulation to study coverage and delay.
A large amount of work has been done in the database community on the topic of epidemic replication protocols [5, 22, 1]. These protocols assume a fully connected network; they are thus not suitable as such in the context we consider. However, the update propagation algorithm we propose for the point-to-point updates between a static node and a smart-tag is similar to the update propagation algorithms from the epidemic replication literature.
Active smart-tags are common place in our daily lives. They are generally used for security (IDs for building entrance) or for tracking (shoe-tags used in running competitions). These smart-tags contain a limited amount of memory and are capable of exchanging data via radio communication. Our implementation uses BlueTag development kits, i.e. alpha versions of Bluetooth based active smart-tags produced by a Danish startup [15]. Bluetooth is an interesting networking technology because of its support for device discovery. Even if we do not focus on energy consumption, we consider that smart-tags should not broadcast data continuously; they should rather transmit data efficiently as soon as they are in the vicinity of a static node. The question is how efficient are the Bluetooth device discovery and connection establishment procedures: how long should a BlueTag remain in the vicinity of a static node for updates to be propagated?
This paper makes the following contributions:

1. We describe an algorithm for propagating updates between a static node and a smart-tag. This algorithm compares version information (i.e., time stamps) and propagates updates to out-of-date versions. This algorithm also accounts for memory limitation on smart-tags.

2. We propose a model for smart-tag based data dissemination. We simulate this model in order to study delay and coverage for the update propagation algorithm we propose.

3. We present performance results using our implementation with an alpha-version of Bluetooth-based smart-tags. In particular we study the performance of the Bluetooth device discovery mechanism and the performance of our update propagation algorithm.

This work is part of the Manatee project at University of Copenhagen where we study Bluetooth-based monitoring applications [19].

2. UPDATE PROPAGATION

We consider a system composed of static nodes and mobile smart-tags that can exchange data when they are in the vicinity of each other (there is no connection among static nodes or among smart-tags [2]). A subset of the static nodes are *data sources*, i.e. they generate 3-tuples ($SourceId$, $Source\ Value$, $SourceTimeStamp$) where $SourceId$ is an identifier of the data source, $SourceValue$ is the data produced by the source (typically a boolean, a float or an integer) and $SourceTimeStamp$ is the point in time at which the data value was generated. Other static nodes are *access points*, whose role is to exchange data with the smart-tags that pass by, store up-to-date data, and possibly display it to users.
Both static nodes and smart-tags maintain a state composed of the data obtained from data sources. This state is basically an array of 2-tuples ($SourceValue$, $SourceTimeStamp$) indexed by the $sourceId$. Note that for a static node which acts as data source id, the smart-tag state is the only data item stored locally, it is regularly updated as measurements are generated [3].
Because connection can potentially be disrupted at any point in time, it is a good policy to exchange the most important data first. An access point updates its state based on the data carried by smart-tags in its vicinity and also updates the smart-tag's state to ensure further data dissemination. By contrast, a data source only updates the smart-tag's state. We present both algorithms in the rest of the section.

2.1 Access Points Updates

The algorithm run by access points to handle update propagation is the following: a connection is established with a smart-tag, the static node first updates its state and then updates the smart-tag state. Here is a pseudocode version of the update propagation algorithm, implemented on access points.

```
loop
{
  begin connection with a smart-tag
  get data from the smart-tag
  update local state
  update smart-tag state
  end connection with the smart-tag
}
```

The static node is responsible for establishing connections with smart-tags. An alternative solution would be to have smart-tag broadcast their state – which does not seem reasonable from the point of view of energy consumption.
The update of the static node is straightforward: it is updated with more up-to-date data obtained from the smart-tag. The $SourceTimeStamp$ is used to compare the version of the data on the smart-tag and on the

[2]In order to minimize the requirements on smart-tags, we assume that smart-tags do not implement update propagation. As a consequence there are no direct communications between smart-tags. Studying an infrastructure where smart-tags can exchange data with each other is a topic for future research.
[3]Note also that we focus on the dissemination of single data items; dealing with time series is a topic for future work.

static node. If the smart-tag version is more recent than the local version, the local version is updated.
The update of the smart-tag is a bit more subtle. Memory limitation might constrain the size of the state on the smart-tags [4]. In most cases, the smart-tag will not be able to store data from all data sources. Static nodes must take this limitation into account when they update the smart-tag state. We distinguish three policies for the access points:

- *STICKY:* The smart-tag keeps the data it carries when it is more recent than the data on the static node. The static node thus updates only the part of the smart-tag state that is outdated; it picks data from data sources in its state following a round-robin policy.

- *ROUND ROBIN:* The static node updates the whole smart-tag state regardless of whether it is outdated or not; it picks data from data sources in its state following a round-robin policy. Note that the static node state is updated before the smart-tag state and as a result, the smart-tag will always carry the latest version of the data.

- *RANDOM:* The static node updates the whole smart-tag state regardless of whether it is outdated or not; it picks data from data sources in its state randomly (avoiding duplicate data sources).

The STICKY policy favors a depth-first dissemination of data (a smart-tag disseminates a data item as far as possible), while ROUND ROBIN and RANDOM favor a breadth-first approach (smart-tags passing by an access point disseminate data from different sources).
In the next section, we analyze how these policies impact coverage and delay.

2.2 Data Sources Updates

In the case of data sources, the latest item generated should be pushed to the smart-tags that pass by. There is no problem if there is enough memory on the smart-tag. In case of memory limitation on the smart-tag, a replacement algorithm needs to be implemented. If the memory of the smart-tag is full, and if the smart-tag does not already store data from the data source it is connected with, then one data item has to be removed from the smart-tag memory and replaced by the data item from the connected data source. This is a classical replacement algorithm. We chose a form of LRU policy, where the data item with the oldest time stamp is replaced.
Note that this buffer replacement policy can be implemented on the smart-tag; in this case the exchange between the data source and the smart-tag is limited to the data source sending its latest data item. If buffer replacement is implemented on the data source then the data source first needs to obtain the smart-tag state in order to apply the buffer replacement algorithm and then send the resulting state back to the smart-tag.

[4]The smart-tags we have used for our implementation have a memory limited to 2Kb, i.e., a state of approximately 100 3-tuples. We further assume that static nodes are able to store data obtained from all data sources. Note that another important form of limitation concerns the amount of data to be exchanged between a static node and a smart-tag. Such a limitation is dictated by energy constraints on both smart-tags and static nodes. Taking the energy cost into account is a topic for future work

3. A MODEL FOR SMART-TAG BASED DATA DISSEMINATION

In this section, we present a model of smart-tag based data dissemination that we simulate in order to illustrate its main characteristics. This simulation is not an in-depth analysis of our data dissemination model: first the experiments we present concern a limited number of parameters (in particular we do not study scalability in terms of data sources), second we make simplifying assumptions concerning the movement of smart-tags and the interaction between smart-tags and static nodes, finally we do not take energy consumption into accounts (energy is an important metrics in the context of sensor networks). A complete study of our data dissemination model is a topic for future work.

3.1 Model Characteristics

Our model of smart-tag based data dissemination can be expressed as follows. The system is composed of a fixed number N of static nodes and a fixed number ST of smart-tags. N_{ds} of the nodes are data sources. Smart-tags are mobile and follow fixed paths between static nodes, i.e., the static nodes are vertices and the paths edges in a connected graph. Each smart-tag moves for a given number of *hops*, where each hop corresponds to a move from one vertex to another following one edge.
We simulated this model using a discrete-event simulator in order to study coverage and delay. In our simulation, the static nodes are arranged following a planar mesh of size $size$, i.e., there are $N = size^2$ nodes; the vertex cardinality is two at the corners, three on the edges and four inside the mesh. This topology is neutral and allows us to focus on other parameters of the model. Data sources are uniformly distributed throughout the mesh. All smart-tags move for a constant number of *hops*. Smart-tags move randomly as follows: a smart-tag follows the edge it just took with a probability of 0.1 and follows all other possible edges with an equal probability. Note also that our simulator assumes that communications between a smart-tag and static nodes always succeed. This is an optimistic assumption because, in practice, a smart-tag might not stay long enough in the vicinity of a static node for the update propagation to complete successfully, or because energy cost considerations force to minimize the transfers between static nodes and smart-tags.
In our experiments, $size$ is 10 and there are 5 data sources. The parameters are thus the number of smart-tags (ST) and the number of hops each smart-tag travels (hop). The idea is to model how data dissemination evolves during a fixed time period, say a day. Each smart-tag moves for a limited number of hops during that period. We choose, for our experiments, a small numbers of hops (20 to 50) and we study how the system scales with the number of smart-tags (20 to 500). Each hop takes one simulation tic. Initially, smart-tags are located at random nodes in the system. The simulation stops when each smart-tag has moved for the specified number of hops.
We ran a few experiments with only one data source, and with enough memory on the smart-tags to carry data from this data source. Our goal was to study the impact of smart-tag movement on coverage.
In a first experiment, we considered that only one data item was generated at the data source. We measured the coverage, defined as the ratio between the number of nodes that have received data from the source and the total number of nodes in the system. We do not