Computer Networks 112 (2017) 294–313

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

A neighbor detection algorithm based on multiple virtual mobile nodes for mobile ad hoc networks

Behnaz Bostanipour*, Benoît Garbinato

Distributed Object Programming Laboratory, Internef Building, University of Lausanne, CH-1015 Lausanne, Switzerland

ARTICLE INFO

Article history:
Received 4 April 2016
Revised 5 November 2016
Accepted 7 November 2016
Available online 9 November 2016

Keywords:
Neighbor detection
Virtual mobile node
MANET
Proximity-based mobile applications
Distributed algorithms
Smartphone

ABSTRACT

We introduce an algorithm that implements a time-limited neighbor detector service in mobile ad hoc networks. The time-limited neighbor detector enables a mobile device to detect other nearby devices in the past, present and up to some bounded time interval in the future. In particular, it can be used by a new trend of mobile applications known as proximity-based mobile applications. To implement the time-limited neighbor detector, our algorithm uses $n = 2^k$ virtual mobile nodes where k is a non-negative integer. A virtual mobile node is an abstraction that is akin to a mobile node that travels in the network in a predefined trajectory. In practice, it can be implemented by a set of real nodes based on a replicated state machine approach. Our algorithm implements the neighbor detector for real nodes located in a circular region. We also assume that each real node can accurately predict its own locations up to some bounded time interval $\Delta_{predict}$ in the future. The key idea of the algorithm is that the virtual mobile nodes regularly collect location predictions of real nodes from different subregions, meet to share what they have collected with each other and then distribute the collected location predictions to real nodes. Thus, each real node can use the distributed location predictions for neighbor detection. We show that our algorithm is correct in periodically well-populated regions. We also define the minimum value of $\Delta_{predict}$ for which the algorithm is correct. Compared to the previously proposed solution also based on the notion of virtual mobile nodes, our algorithm has two advantages: (1) it tolerates the failure of one to all virtual mobile nodes; (2) as n grows, it remains correct with smaller values of $\Delta_{predict}$. This feature makes the real-world deployment of the neighbor detector easier since with the existing prediction methods, location predictions usually tend to become less accurate as $\Delta_{predict}$ increases. We also show that the cost of our algorithm (in terms of communication) scales linearly with the number of virtual mobile nodes.

1. Introduction

The growing adoption and usage of mobile devices and particularly smartphones has caused the emergence of a new trend of distributed applications known as *Proximity-Based Mobile* (*PBM*) applications [3–6]. These applications enable a user to interact with others in a defined range and for a given time duration e.g., for social networking (WhosHere [52], LoKast [35], iGroups [25], LocoPing [34]), gaming (local multiplayer apps [36]) and driving (Waze [51]).

Discovering who is nearby is one of the basic requirements of PBM applications. It is the preliminary step for further interactions between users. It also enables users to extend their social network from the people that they know to the people that they might not know but who are in their proximity. For instance, with the social networking applications such as WhosHere [52] or LoKast [35], a user first discovers others in her proximity and then decides to view their profiles, start a chat with them or add them as friends. The discoverability, however, may not always be limited to the current neighbors. For instance, with the social networking applications such as iGroups [25] or LocoPing [34], a user can discover others who were in her vicinity during a past event (e.g., concert, tradeshow, wedding) or simply during a past time interval (e.g., the past 24 h). One can also think of applications that provide the user with the list of people who will be in her proximity up to some time interval in the future and thus create the potential for new types of social interactions [5].

In this paper, we present an algorithm that implements the *time-limited neighbor detector* service. This service enables a device to discover the set of its neighbors in the past, present and up to

* Corresponding author.
E-mail addresses: behnaz.bostanipour@unil.ch (B. Bostanipour), benoit.garbinato@unil.ch (B. Garbinato).

http://dx.doi.org/10.1016/j.comnet.2016.11.004

some bounded time interval in the future in a mobile ad hoc network (MANET). It was first introduced in our previous work [5] to capture the requirements of neighbor detection in PBM applications.

Our algorithm implements the time-limited neighbor detector using $n = 2^k$ virtual mobile nodes where k is a non-negative integer.[1] A virtual mobile node is an abstraction that was already introduced in the literature and used for tasks such as routing or collecting data in MANETs [14,15]. It is akin to a mobile node that travels in the network in a predefined trajectory known in advance to all nodes. In practice a virtual mobile node is emulated by a set of real nodes in the network using a replicated state machine approach.

Our algorithm implements the neighbor detector for real nodes located in a circular region. We also assume that each real node can accurately predict its own locations up to some bounded time interval $\Delta_{predict}$ in the future. Thus, the region is divided into n equal subregions and each subregion is associated with one virtual mobile node. Each virtual mobile node regularly collects the location predictions from the real nodes in its subregion and meets other virtual mobile nodes to share what it has collected with them. After the sharing, each virtual mobile node has the location predictions collected from the entire region, which it distributes to the real nodes in its subregion. In this way, each real node can find its neighbors at current and future times based on the collected location predictions that it receives from a virtual mobile node. It can also store the collected location predictions so it can be queried about its past neighbors.

Main contributions. The main contributions of this paper are as follows. We introduce an algorithm that implements the time-limited neighbor detector service using $n = 2^k$ virtual mobile nodes where k is a non-negative integer. To guarantee the coordination between the virtual mobile nodes, we define a set of explicit properties for their trajectory functions and we show how such trajectory functions can be computed. We prove the correctness of the algorithm under certain conditions. In particular, we show that our algorithm is correct for a category of executions, called *nice executions*, which basically correspond to the executions of the algorithm in periodically well-populated regions such as main squares in a downtown area. We also define the minimum value of $\Delta_{predict}$ for which the algorithm is correct in different cases of nice executions.

This work relies on our previous work [5] for the general idea of using virtual mobile nodes and location predictions to implement the time-limited neighbor detector. However, contrary to the algorithm in [5] which uses only a single virtual mobile node and does not tolerate its failure, our algorithm can use multiple virtual mobile nodes and can tolerate the failure of one to all virtual mobile nodes. Due to the use of multiple virtual mobile nodes, our algorithm has a feature which did not exist in the previous solution: as the number of virtual mobile nodes grows, our algorithm remains correct with smaller values of $\Delta_{predict}$. This feature makes the real-world deployment of the neighbor detector easier. In fact, although there exist different approaches to predict the future locations of a real node, usually predictions tend to become less accurate as $\Delta_{predict}$ increases. We show that the cost of our algorithm (in terms of communication) scales linearly with the number of virtual mobile nodes. We also propose a set of optimizations which can be used for the real-world deployment of our algorithm.

To the best of our knowledge, this is the first work that uses multiple virtual mobile nodes to implement a neighbor detector service in MANETs. Moreover, this is the first work that defines a set of explicit properties for the trajectory functions of the virtual mobile nodes to guarantee the coordination between them.

Road map. The remainder of the paper is as follows. In Section 2, we describe our system model and introduce some definitions. In Section 3, we present a neighbor detector service for MANETs in two variants: the *perfect* variant, which corresponds to the ideal case of neighbor detection and is rather impractical and the *time-limited* variant, for which we propose an implementation in this paper. In Section 4, we present the implementation of the time-limited variant of the neighbor detector service based on virtual mobile nodes. In order to do so, we first describe what a virtual mobile node is and how it can be used for the implementation of the time-limited neighbor detector. We then add n virtual mobile nodes to the system model. Each virtual mobile node has a so called *scan path* through which it travels in its subregion. Thus, we define the properties of this path and we show how it can be computed in order to be useful for our algorithm. We then introduce a round-based algorithm that implements the time-limited neighbor detector in the new system model and prove the correctness of the algorithm under certain conditions. As we show in the proof, the algorithm can tolerate the failure of one to all virtual mobile nodes for a category of executions, called *nice executions*, which basically correspond to the executions of the algorithm in periodically well-populated regions. We also define the minimum value of $\Delta_{predict}$ for which the algorithm is correct in different cases of nice executions. Then, we show the evolution of this value as n grows. Based on this evolution, we deduce that as the number of virtual mobile nodes grows the algorithm requires smaller values of $\Delta_{predict}$ to correctly implement the time-limited neighbor detector. In Section 5, we discuss two topics related to the performance of the algorithm, namely its scalability with respect to the number of virtual mobile nodes and the optimizations which can improve its performance. In particular, we show that the communication cost of the algorithm, defined as the number of message broadcasts per round, has a complexity of $\mathcal{O}(n)$. In Section 6, we discuss the related work and in Section 7, we conclude and discuss future work. The paper has also an Appendix A, which is devoted to finding an upper bound for the scan path length of a virtual mobile node. This upper bound is used (directly or indirectly) in Sections 4.3, 4.6 and 5.1 to find other results.

2. System model and definitions

We consider a mobile ad-hoc network (MANET) consisting of a set P of processes that move in a two dimensional plane. A process abstracts a mobile device in a PBM application.[2] We use the terms *process*, *node* and *real node* interchangeably. Each process has a unique identifier. Processes can move on any continuous path, however there exists a known upper bound on their motion speed. A process is prone to *crash-reboot* failures: it can fail and recover at any time, and when the process recovers, it returns to its initial state. A process is *correct* if it never fails. Since we do not consider *Byzantine* behaviors, the information security and privacy issues are beyond the scope of this paper.

We assume the existence of a discrete global clock, i.e., the range T of the clock's ticks is the set of non-negative integers. We

[1] The present work is an extension of the work published as a short conference paper in [7].

[2] There are two main reasons behind our choice of a MANET as the underlying network architecture. Firstly, MANETs seem to be the most natural existing technology to enable PBM applications. In fact, similar to PBM applications, in a MANET two nodes can communicate if they are within a certain distance of each other (to have radio connectivity) for a certain amount of time. Secondly, mobile devices are increasingly equipped with ad hoc communications capabilities (e.g., WiFi in ad hoc mode or Bluetooth) which increases the chance of MANETs to be one of the future mainstream technologies for PBM applications [6].