Information Processing Letters 133 (2018) 39–43

Contents lists available at ScienceDirect

Information Processing Letters

www.elsevier.com/locate/ipl

On deterministic rendezvous at a node of agents with arbitrary velocities

Sébastien Bouchard[a], Yoann Dieudonné[b,*], Andrzej Pelc[c,1], Franck Petit[a,2]

[a] Sorbonne Universités, UPMC Université Paris 06, CNRS, INRIA, LIP6 UMR 7606, Paris, France
[b] MIS Lab., Université de Picardie Jules Verne, France
[c] Département d'informatique, Université du Québec en Outaouais, Gatineau, Québec J8X 3X7, Canada

ARTICLE INFO

Article history:
Received 26 September 2016
Received in revised form 16 November 2017
Accepted 18 January 2018
Available online 31 January 2018
Communicated by P. Wong

Keywords:
Deterministic rendezvous
Algorithms
Mobile agent
Velocity

ABSTRACT

We consider the task of rendezvous in networks modeled as undirected graphs. Two mobile agents with different labels, starting at different nodes of an anonymous graph, have to meet. This task has been considered in the literature under two alternative scenarios: *weak* and *strong*. Under the weak scenario, agents may meet either at a node or inside an edge. Under the strong scenario, they have to meet at a node, and they do not even notice meetings inside an edge. Rendezvous algorithms under the strong scenario are known for synchronous agents. For asynchronous agents, rendezvous under the strong scenario is impossible even in the two-node graph, and hence only algorithms under the weak scenario were constructed. In this paper we show that rendezvous under the strong scenario is possible for agents with asynchrony restricted in the following way: agents have the same measure of time but the adversary can impose, for each agent and each edge, the speed of traversing this edge by this agent. The speeds may be different for different edges and different agents but all traversals of a given edge by a given agent have to be at the same imposed speed. We construct a deterministic rendezvous algorithm for such agents, working in time polynomial in the size of the graph, in the length of the smaller label, and in the largest edge traversal time.

© 2018 Elsevier B.V. All rights reserved.

1. Introduction

The background. We consider the task of rendezvous in networks modeled as undirected graphs. Two mobile entities, called agents, have different positive integer labels, start from different nodes of the network, and have to meet. Mobile entities may represent software agents in a communication network, or physical mobile robots, if the network is a labyrinth or a cave, or if it consists of corridors of a building. The reason to meet may be to exchange previously collected data or ground or air samples, or to split work in a future task of network exploration or maintenance.

The task of rendezvous in networks has been considered in the literature under two alternative scenarios: *weak* and *strong*. Under the weak scenario [6,7,10], agents may meet either at a node or inside an edge. Under the strong scenario [8,13,17], they have to meet at a node, and they do not even notice meetings inside an edge. Each of these scenarios is appropriate in different applications. The weak scenario is suitable for physical robots in a network of corridors, while the strong scenario is needed for software agents in computer networks.

[*] Corresponding author.
E-mail addresses: sebastien.bouchard@lip6.fr (S. Bouchard), yoann.dieudonne@u-picardie.fr (Y. Dieudonné), pelc@uqo.ca (A. Pelc), Franck.Petit@lip6.fr (F. Petit).

[1] Supported in part by NSERC discovery grant 8136 – 2013 and by the Research Chair in Distributed Computing of the Université du Québec en Outaouais.

[2] Supported by the ANR project ESTATE ANR-16-CE25-0009-03.

https://doi.org/10.1016/j.ipl.2018.01.003
0020-0190/© 2018 Elsevier B.V. All rights reserved.

Rendezvous algorithms under the strong scenario are known for synchronous agents, where time is slotted in rounds, and in each round each agent can either wait at a node or move to an adjacent node. For asynchronous agents, where an agent decides to which neighbor it wants to move but the adversary totally controls the walk of the agent and can arbitrarily vary its speed, rendezvous under the strong scenario is impossible even in the two-node graph, and hence only algorithms under the weak scenario were constructed.

However, due to the fact that the strong scenario is appropriate for software agents in computer networks, and that such agents are rarely synchronous, it is important to design rendezvous algorithms under the strong scenario, restricting the asynchrony of the agents as little as possible. This is the aim of this paper. We consider mobile agents with asynchrony restricted as follows: agents have the same measure of time but the adversary can impose, for each agent and each edge, the speed of traversing this edge by this agent. The speeds may be different for different edges and different agents but all traversals of a given edge by a given agent have to be at the same imposed speed. We are interested in deterministic rendezvous algorithms for such agents.

The model.

The network. The network is modeled as a simple undirected connected graph. As in the majority of papers on rendezvous, we seek algorithms that do not rely on the knowledge of node labels, and we assume that the underlying graph is anonymous. Designing such algorithms is important because even when node labels exist, nodes may refuse to reveal them, *e.g.*, due to security or privacy reasons. It should be also noted that, if nodes had distinct labels, agents might explore the graph and meet in the smallest node, hence gathering would reduce to graph exploration. On the other hand, we make the assumption, again standard in the literature of the domain, that edges incident to a node v have distinct labels in $\{0,\dots,d-1\}$, where d is the degree of v. Thus every undirected edge $\{u,v\}$ has two labels, which are called its *port numbers* at u and at v. Port numbers are visible to the agents. Port numbering is *local*, *i.e.*, there is no relation between port numbers at u and at v. Note that in the absence of port numbers, edges incident to a node could not be distinguished by agents and thus rendezvous would be often impossible, as the adversary could prevent an agent from traversing some edge incident to the current node. Also, the above mentioned concerns of security and privacy that may prevent nodes from revealing their labels, do not apply to port numbers.

The agents. Agents A_1 and A_2 start at arbitrary different nodes of the graph. They are placed at their starting nodes at the beginning. They cannot mark visited nodes or traversed edges in any way. The adversary wakes up the agents at possibly different times. Agents do not know the topology of the graph nor any bound on its size. They have clocks ticking at the same rate. The clock of each agent starts at its wakeup, and at this time the agent starts executing the algorithm. When an agent enters a node, it learns its degree and the port of entry. We assume that the memory of the agents is unlimited: from the computational point of view they are modeled as Turing machines.

The adversary. The adversary assigns different positive integer labels to both agents. Each agent knows a priori only its own label. Both agents execute the same deterministic algorithm whose parameter is the label of the agent. Moreover, for each edge e of the graph, the adversary assigns two positive reals: $t_1(e)$ and $t_2(e)$. During the execution of an algorithm, an agent can wait at the currently visited node for a time of its choice, or it may choose a port to traverse the corresponding edge e. In the latter case, agent A_r traverses this edge in time $t_r(e)$, getting to the other end of the edge after this time. This modelling permits a lot of asynchrony: agents can have different velocities when traversing different edges, and an agent slower in one edge can be faster in another edge. This is motivated by the fact that congestion and bandwidth of different edges may be different, and that each of the agents can have a different traversing priority level on different edges. In particular, this general scenario includes the model of agents walking at possibly different constant velocities, that was used in [9] for the task of approach in the plane.

The *time* of a rendezvous algorithm is the worst-case time between the wakeup of the earlier agent and the meeting at a node.

Our results. We construct a deterministic rendezvous algorithm working for arbitrary graphs under the strong scenario. Our algorithm works in time polynomial in n, ℓ and τ, where n is the number of nodes of the graph, ℓ is the logarithm (*i.e.*, the length) of the smaller label, and τ is the maximum of all values $t_1(e)$ and $t_2(e)$ assigned by the adversary, over all edges e of the graph.

Related work. A survey of randomized rendezvous in various scenarios can be found in [1]. Deterministic rendezvous in networks was surveyed in [15]. In many papers rendezvous was considered in a geometric setting: an interval of the real line, see, *e.g.*, [4,12], or the plane, see, *e.g.*, [2,3].

For deterministic rendezvous in networks, attention concentrated on the study of the feasibility of rendezvous, and on the time required to achieve this task, when feasible. For example, deterministic rendezvous with agents equipped with tokens used to mark nodes was considered, *e.g.*, in [14]. Deterministic rendezvous of two agents that cannot mark nodes but have unique labels was discussed in [8,13,17]. All these papers were concerned with the time of rendezvous in arbitrary graphs. In [8] the authors showed a rendezvous algorithm polynomial in the size of the graph, in the length of the shorter label and in the delay between the starting times of the agents. In [13,17] rendezvous time was polynomial in the first two of these parameters and independent of the delay. All the above papers assumed that agents are synchronous, and used the scenario called *strong* in the present paper.

Several authors investigated asynchronous rendezvous in the plane [5,11] and in network environments [6,7,10]. In the latter scenario, the agent chooses the edge which it decides to traverse but the adversary totally controls the walk of the agent inside the edge and can arbitrarily vary

Download English Version:

https://daneshyari.com/en/article/6874214

Download Persian Version:

https://daneshyari.com/article/6874214

Daneshyari.com