Journal of Applied Logic 20 (2017) 1–31

Contents lists available at ScienceDirect

Journal of Applied Logic

www.elsevier.com/locate/jal

Epistemic protocols for dynamic gossip

Hans van Ditmarsch [a], Jan van Eijck [b], Pere Pardo [c,*], Rahim Ramezanian [d], François Schwarzentruber [e]

[a] *LORIA, CNRS, University of Lorraine, France*
[b] *CWI and ILLC, University of Amsterdam, Netherlands*
[c] *Institut für Philosophie II, Ruhr-Universität Bochum, Germany*
[d] *Sharif University of Technology, Iran*
[e] *ENS Rennes, IRISA, France*

A R T I C L E   I N F O

Article history:
Received 14 April 2016
Accepted 28 November 2016
Available online 5 December 2016

Keywords:
Gossip
Dynamic networks
Distributed protocols
Communication
Epistemic logic
Agents

A B S T R A C T

A gossip protocol is a procedure for spreading secrets among a group of agents, using a connection graph. In each call between a pair of connected agents, the two agents share all the secrets they have learnt. In dynamic gossip problems, dynamic connection graphs are enabled by permitting agents to spread as well the telephone numbers of other agents they know. This paper characterizes different distributed epistemic protocols in terms of the (largest) class of graphs where each protocol is successful, i.e. where the protocol necessarily ends up with all agents knowing all secrets.

1. Introduction

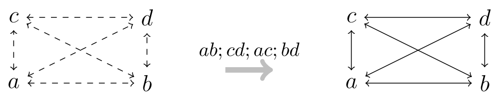
Gossip protocols are procedures for spreading secrets among a group of agents, using a connection graph.[1] This connection graph denotes which agents know the telephone numbers of who, and so which telephone calls can be made. During each call, say ab from a to b, the two agents a and b share all the secrets they know at the time of the call (initially, each agent only knows its own secret). In the original set-up, where a totally connected graph was assumed, the main question was to find a sequence of calls making all agents knew all secrets (henceforth, successful) with a minimal number of calls; this minimum length turns out to be $2n - 4$ in a totally connected graph with $n > 3$ agents, see Tijdeman [22] or Hurkens [17]. Consider the totally connected graph with four agents, and the successful call sequence $ab; cd; ac; bd$ in the following

* Corresponding author.
E-mail addresses: hans.van-ditmarsch@loria.fr (H. van Ditmarsch), jan.van.eijck@cwi.nl (J. van Eijck), pere.pardoventura@mail.ruhr-uni-bochum.de (P. Pardo), rahim_ramezanian@mehr.sharif.ir (R. Ramezanian), francois.schwarzentruber@ens-rennes.fr (F. Schwarzentruber).

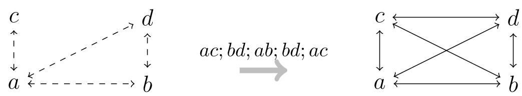
[1] For an example of application, consider the problem of *scheduling a meeting* for a group of agents. The *secret* of each agent is its time availability for possible meetings.

http://dx.doi.org/10.1016/j.jal.2016.12.001
1570-8683/

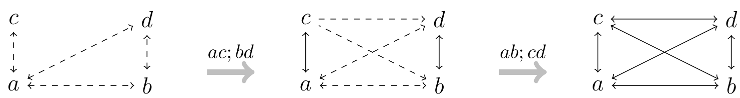
picture (where dashed arrows denote knowledge of telephone number, and solid arrows denote who knows whose secrets; note we do not draw the reflexive edges corresponding to the fact that agents always know their own secret and telephone number).



Graphs that are not totally connected were also studied by different authors; [14] is a survey of results on the gossip problem, including the following minimum bounds on the length of successful sequences: for any (undirected) connected graph, there is a successful call sequence of length $2n - 3$ for $n > 3$ agents; moreover, if the graph contains a 4-cycle (as in totally connected graphs), then the minimum is again $2n-4$ calls. Consider for example the following connection graph:



In the connection graph above, the call sequence $ac; bd; ab; bd; ac$ can be improved if we permit the agents to exchange, during each call, both the secrets and the telephone numbers. In this setting, called *dynamic gossip* in some previous work [24], we can recover the $2n - 4$ minimum length for the above gossip graph.



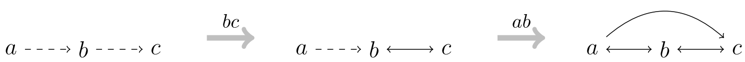
In all of the examples above, a way to ensure that minimum length solutions will be found is to assume a central authority who, knowing the initial network topology, designs the desired call sequence and communicates it to the agents. In distributed computing, in contrast, one is interested in procedures that do not need outside regulation, even if this comes at the price of non-optimal call sequences. Among distributed protocols that might find the above solutions one can mention

Any Call While not every agent knows all secrets, randomly select a pair xy such that x knows y's number and let x call y.

or, with less redundancy (since the sequence $ac; ac; bd; ab; cd$ is not permitted), the protocol

Learn New Secrets While not every agent knows all secrets, randomly select a pair xy such that x knows y's number but not her secret and let x call y.

The following execution $bc; ab$ of the Learn New Secrets protocol is unsuccessful in the next graph:



One can observe that after the second call ab, in the above sequence $bc; ab$, agents a and b know that it surely makes sense to make a new call ac or bc, since agent c would then learn something, namely, the initial secret from a. One protocol to address this is

Known Information Growth While not every agent knows all secrets, randomly select a pair xy such that x knows y's number and there is a z such that x knows that either x or y doesn't know z's secret; let x call y.