Discrete Applied Mathematics 201 (2016) 114–129

Contents lists available at ScienceDirect

Discrete Applied Mathematics

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

Network decontamination under m-immunity[☆]

Paola Flocchini[a], Fabrizio Luccio[b], Linda Pagli[b], Nicola Santoro[c,*]

[a] University of Ottawa, Canada

[b] Università di Pisa, Italy

[c] School of Computer Science, Carleton University, K1S 5B6 Ottawa, Canada

ARTICLE INFO

Article history:
Received 28 January 2014
Received in revised form 18 July 2015
Accepted 20 July 2015
Available online 18 August 2015

Keywords:
Network decontamination
Connected graph search
Monotone algorithms
Immunity
Meshes
Tori
Trees

ABSTRACT

We consider the problem of decontaminating an infected network using as few mobile cleaning agents as possible and avoiding recontamination. After a cleaning agent has left a vertex v, this vertex will become recontaminated if m or more of its neighbors are infected, where $m \geq 1$ is a threshold parameter of the system indicating the local immunity level of the network. This *network decontamination* problem, also called *monotone connected graph search* and *intruder capture*, has been extensively studied in the literature when $m = 1$ (no immunity).

In this paper, we extend these investigations and consider for the first time the network decontamination problem when the parameter m is an arbitrary integer value $m \geq 1$. We direct our study to widely used interconnection networks, namely meshes, tori, and trees. For each of these classes of networks, we present decontamination algorithms with threshold m; these algorithms work even in asynchronous setting, either directly or with a simple modification requiring one additional agent. We also establish general lower bounds on the number of agents necessary for decontamination with immunity m; these bounds are tight in the case of trees, while large gaps still exist in the case of meshes and tori.

1. Introduction

1.1. The framework

Parallel and distributed computing systems are designed around interconnection networks. As the size, the complexity, and the importance of a system increase, the presence of malicious threats cannot be avoided. Such threats may be brought by *intruders* that travel through the network and infect any visited site, as for example a virus. The focus of this paper is on counteracting such a threat by a team of mobile *cleaning agents* (or simply *agents*) that traverse the network decontaminating the visited sites [15].

The team of mobile agents enter the network, viewed as a simple undirected graph, at a single vertex, called *homebase*. An agent located at a vertex v can move to any of the neighboring nodes of v, decontaminating it with its presence; upon departure of the (last) agent, a vertex can become re-contaminated if a sufficient number of its neighbors are contaminated. The goal is to decontaminate the whole network using as small a team of cleaning agents as possible avoiding any

[☆] Some of these results have been presented at the 23rd IEEE International Parallel and Distributed Processing Symposium and at the 8th International Conference on Algorithms and Complexity.

[*] Corresponding author. Tel.: +1 613 520 4333; fax: +1 613 520 4336.
E-mail addresses: flocchin@site.uottawa.ca (P. Flocchini), luccio@di.unipi.it (F. Luccio), pagli@di.unipi.it (L. Pagli), santoro@scs.carleton.ca (N. Santoro).

http://dx.doi.org/10.1016/j.dam.2015.07.020
0166-218X/

recontamination. This *network decontamination* problem, known also as *monotone connected graph search* and as *intruder capture*, has been extensively studied in the literature (e.g., see [1,2,6,12,11,13,16,17,19,25,24,28,33]).

The re-contamination process is regulated by a parameter m: if no agent is there, a decontaminated vertex v will become re-contaminated if m or more of its neighbors are infected. The parameter m indicates the local *immunity* level of the network. The number of agents necessary to decontaminate the entire network is clearly a function of m and of the basic parameters of the given network.

In the literature the problem has generally been studied only when $m = 1$, that is the presence of a single infected neighbor recontaminates a disinfected node with no agents. This assumption corresponds to a system without any immunity, and the problem has been investigated under such assumption for a variety of network classes, including trees, hypercubes, meshes, tori, outerplanar graphs, chordal graphs, etc. (e.g., see [3,9,12,11,13,17,28]).

The assumption $m = 1$ is quite restrictive. In fact, to enhance reliability, many systems employ *threshold* rules at each site, for example performing voting among various copies of crucial data between neighbors at each step [31]. Indeed, threshold schemes are used for consistency resolution protocols in distributed database management; data consistency protocols in quorum systems; mutual exclusion algorithms; key distribution in security; reconfiguration under catastrophic faults in system level analysis; and computational models in discrete-time dynamical systems. This leads to consider systems with a higher level of resistance to viral threats, where a vertex can be recontaminated after an agent has left only if a threshold $m > 1$ of its neighbors are infected. In turn, this opens the research investigation of the general decontamination problem when the threshold indicating the level of *local immunity* is a global parameter $m \geq 1$. This is precisely the question we address in this paper.

1.2. Main contributions

We investigate the decontamination problem for arbitrary $m \geq 1$. We focus on three common classes of interconnection networks: *Meshes*, *Tori*, and *Trees*; for each network G in those classes, we establish bounds on the number of agents necessary to decontaminate G with threshold m. The upper bound proofs are constructive, and some of our decontamination protocols are shown to be optimal. More precisely:

- We first consider d-dimensional *meshes* with $N = n_1 \times n_2 \times \cdots \times n_d$ vertices, where $d \geq 2$; w.l.g., let $2 \leq n_1 \leq n_2 \leq \cdots \leq n_d$. We prove that for each such mesh one agent suffices for $m \geq d$, and $n_1 \times n_2 \times \cdots \times n_{d-m}$ agent suffice for $1 \leq m < d$ by exhibiting a solution algorithm that uses these many agents. We also establish a general lower bound, and show it is tight for $m = d - 1$.
- We then extend the analysis to d-dimensional *toroidal meshes* M, with $N = n_1 \times n_2 \times \cdots \times n_d$ vertices, where $d \geq 2$; w.l.g., let $2 \leq n_1 \leq n_2 \leq \cdots \leq n_d$. We prove that $2^m \times n_1 \times n_2 \times \cdots \times n_{d-m}$ agents suffice for $1 \leq m \leq d-1$, and 2^{2d-m} agents suffice for $d \leq m \leq 2d$.

  Note that with growing m more infected neighbors must be present to re-contaminate a vertex, so the number of agents must decrease if m grows. This is immediately clear in our protocol for $d \leq m \leq 2d$. For $1 \leq m \leq d - 1$ the number of agents grows with the term 2^m but decreases more rapidly with the number of n_i present in the bound, under the obvious hypothesis that all n_i are greater than 2.

  We also establish a general lower bound, and show it is tight for some values of d and m.
- Finally we consider the family of *trees*. Unlike the case of meshes and tori, the number of needed cleaning agents may be different for different trees of the same size. For every tree and any value of $m \geq 1$ we determine a lower bound to the number of agents, and we then prove that this number is also sufficient by presenting a simple decontamination protocol using precisely those many agents.

The algorithms for meshes and tori are established in a quasi-synchronous setting: agents operate in synchronized steps, but operations within a step (e.g., movements) are not necessarily synchronized. The algorithms can be extended to completely asynchronous settings with simple modifications to their structure and the use of one single extra agent, as in [12,11,13]. The proposed algorithms for trees work directly in asynchronous settings.

Although not the main concern of this paper, we also consider the number of moves performed by an optimal-size team of agents as a function of the parameter m. We prove that, for all three families of graphs, all the solution protocols we have presented are optimal, in order of magnitude, with respect to the number of moves.

The paper is organized as follows. In Section 2, we introduce the model and basic properties. In Sections 3–5 we present decontamination algorithms for meshes, toroidal meshes, and trees, with the upper and lower bounds on the numbers of agents and moves. In the concluding Section 6 we discuss extension of our studies.

1.3. Previous work

The network decontamination problem was surprisingly introduced in speleology [7], and then has been extensively studied in the field of *graph searching* (e.g., see [4,5,26]). The problem also arises in graph pebbling [22], as a pursuit-evasion game [27,30], and in VLSI design [21]. An important aspect of these studies is that there are always *monotone* solutions that use a minimum number of agents, that is protocols that avoid recontamination of vertices after they have been decontaminated [5,23]. Although formally very similar to ours, all these investigations assume that the decontaminating