Theoretical Computer Science 627 (2016) 1–12

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

The impact of dynamic events on the number of errors in networks

Christian Glacet [a,*,1], Nicolas Hanusse [b,2], David Ilcinkas [b,2]

[a] CNR, IEIIT, Turin, Italy
[b] CNRS & Univ. Bordeaux, LaBRI, UMR 5800, F-33400 Talence, France

ARTICLE INFO

Article history:
Received 21 January 2015
Received in revised form 21 January 2016
Accepted 12 February 2016
Available online 17 February 2016
Communicated by M. Latapy

Keywords:
Graph
Network
Dynamics
Errors
Routing information
Distance changes

ABSTRACT

In order to achieve routing in a graph, nodes need to store routing information. In the case of shortest path routing, for a given destination, every node has to store an *advice* that is an outgoing link toward a neighbor. If this neighbor does not belong to a shortest path then the advice is considered as an error and the node giving this advice will be qualified a *liar*. This article focuses on the impact of graph dynamics on the advice set for a given destination. More precisely we show that, for a weighted graph G of diameter D with n nodes and m edges, the expected number of errors after $\mathcal{M}$ edge deletions is bounded by $\mathcal{O}\left(n \cdot \mathcal{M} \cdot {}^{D}/_{m}\right)$. We also show that this bound is tight when $\mathcal{M} = o(n)$. Moreover, for $\mathcal{M}'$ node deletions, the expected number of errors is $\mathcal{O}\left(\mathcal{M}' \cdot D\right)$. Finally we show that after a single edge addition the expected number of liars can be $\Theta(n)$ for some families of graphs.

1. Introduction

Everyone has already faced the problem of reaching a destination in an uncertain network. This is typically the case whenever one is in an unknown city, without a map, and aims at reaching, let us say, the closest cash machine. The only thing one can do is ask people in the street for some information on the direction. Unfortunately, there is no evidence that all the information one gets is reliable.

Nowadays, in a communication network, a corresponding situation can occur. Let us consider the routing task. Due to their dynamics (change of topology, time required to update local information) and their large-scale size, current networks are not immune to faults and crashes. It is no more realistic to blindly trust the data stored locally at each node. For instance, the Border Gate Protocol (BGP) used on the Internet to route messages between autonomous systems implicitly assumes that some paths are known to reach any target. Ideally, these paths are as short as possible. Unfortunately, many messages do not reach their destination because no paths are temporally known although some paths could exist.

In the following, for a given target t, we informally refer to a *liar* as a node containing bad information about the location of t. A series of papers [11,10,9] tackles the problem of locating a target (node, resource, data, ...) in presence of liars.

* Corresponding author.
E-mail address: christian.glacet@gmail.com (C. Glacet).
[1] Partially supported by the ERCIM fellowship program.
[2] This work was partially funded by the ANR project DISPLEXITY (ANR-11-BS02-014). This study has been carried out in the frame of “the Investments for the future” Programme IdEx Bordeaux – CPU (ANR-10-IDEX-03-02).

http://dx.doi.org/10.1016/j.tcs.2016.02.012
0304-3975/

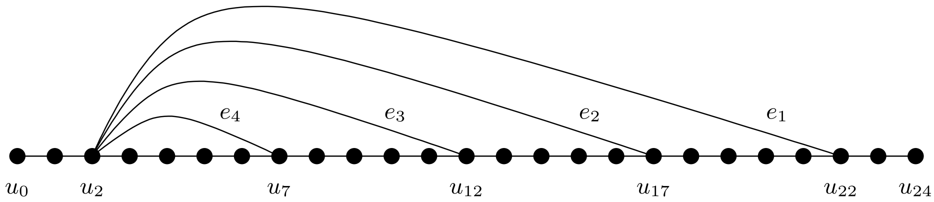

Fig. 1. Lower bound example on the impact of dynamics on the diameter from [4].

A first model was introduced by Kranakis and Krizanc [15]. They designed algorithms for searching in distributed networks having the ring or the torus topology, when a node has a constant probability of being a liar. A more realistic model was proposed by Hanusse et al. [11]: the number of liars is a parameter k and during a routing query, the information stored at every node is unchanged. The main performance measure is the number of edge traversals during a request. Several algorithms, either generic or dedicated to some topologies, and bounds are presented in [11,10,9] and are typically of the form $\mathcal{O}\big(d + k^{\mathcal{O}(1)}\big)$ (for path, grids, expanders, ...) or $\mathcal{O}\Big(k^3 \log^3 n\Big)$ for bounded degree graphs, d being the distance between the source and the target.

In these papers, there is an implicit assumption: the number of liars is small. Our goal is to evaluate whether this is realistic or not. Starting from a network without any liar, we aim at estimating bounds on the number of liars obtained after few changes of topology. It turns out that this problem is related to the problem of estimating the number of distance changes after few edge/node deletions or insertions. We will show that the expected number of liars created by random nodes/edges removals is proportional to the diameter of the graph and its average degree. Sections 2 and 4 will respectively cover the related works and the presentation of our results, while sections 5, 6 and 7 will show the details of the proofs. Section 5 contains general results, Sections 6 and 7 respectively concern deletions and additions."

2. Related works

The influence of topology changes in graphs has been studied in several works. In [4,18], it has been proven that, in the worst case scenario, the deletion of $\mathcal{M}$ edges within a graph of diameter D induces a multiplication of the diameter by a factor $O(\mathcal{M}+1)$. A lower bound of $(\mathcal{M}+1)D - 2\mathcal{M} + 2$ can be obtained, for odd D using the very simple example given in [4] and showed in Fig. 1. In this example, the graph G has diameter $D(G) = 6$ and $n = 25$ nodes, the deletion of edges e_1, e_2, e_3 and e_4 induces a graph G' of diameter $D(G') = n - 1 = 24 = (\mathcal{M}+1) \cdot D(G) - 2\mathcal{M} + 2$. In other words, every edge deletion adds roughly $D(G) = 6$ units to the diameter of the graph.

Our work is also related to the computation of the most vital node of a shortest path [17], that is the node whose removal results in the largest increase of the distance for a given pair of source/target, and the Vickrey pricing of edges [12]. More precisely, the latter article shows a centralized algorithm that allows to find the most important node for a user that would like to send data over a given graph. This algorithm runs in time $\mathcal{O}(m + n \log n)$ with a working memory of $\mathcal{O}(m)$.

Recently, some works on *dynamic* data structures for shortest paths/distance computation problems have been proposed. By dynamic, we mean that the data structures can tolerate some topology changes in a given network. A dynamic network model defines how the underlying graph changes/evolves over time. More precisely, the following types of models are usually considered:

- The most general model is the model of unconstrained *Evolving graphs* or *Time Varying Graphs*, introduced by Ferreira in [8]: An evolving graph $\mathcal{G}$ is based on a static graph $G = (V, E)$ called *underlying graph*. The state of the graph $\mathcal{G}$ at a given time t, $G_t = (V, E_t)$, is given by a presence function which determines for every edge of E if it belongs to E_t or not. Thus, for every t, the graph G_t is a sub-graph of G. Let remark that for any two given times t and t' the state of the graphs G_t and $G_{t'}$ can be very different, which means that the network can vary at any speed.
  This type of unconstrained evolving graphs makes however most of the static graph problems unsolvable. This is why weaker models are most commonly considered. For example, in an unconstrained evolving graph, it is impossible to broadcast a piece of information since the graph is not necessary connected over time, i.e. the union of the G_t is not a connected graph. Different variants of this model can be found in the survey made by Santoro et al. in [2].
- However, in many studies the dynamic model studied is even more constrained. A very common model in the routing community is the *failure model*. Intuitively, it is considered in this model that, starting from an initial graph, after $\mathcal{M}$ additions/deletions of nodes/edges the graph stays stable for a sufficiently long period of time. This model is widely used in self-stabilization and in routing or distance oracle problems (also referred as *forbidden-set* problems).

In the case of the failure model, the most naive solution to allow routing is to recompute every routing table after some dynamic events occur. Two dynamic and centralized algorithms that perform a global re-computation are presented by Demetrescu and Italiano [6] and Thorup [19]. These two algorithms are the fastest known and allow to update every routing tables within amortized time of $\mathcal{O}\big(n^2 \,\mathrm{polylog}\,(n)\big)$ per atomic topological modification.

However, in the failure model, it is not always necessary to recompute every shortest path in order to guarantee that routing is possible. In fact, several other works deal with this problem, and propose for example distributed structures that