

Distributed detection of topological changes in communication networks [★]

Riccardo Lucchese^{*} Damiano Varagnolo^{**}
Karl H. Johansson^{***}

^{*} *Department of Information Engineering, University of Padova, Italy.*

^{**} *Department of Computer Science, Electrical and Space Engineering,
Luleå University of Technology, Luleå, Sweden*

^{***} *School of Electrical Engineering, KTH Royal Institute of
Technology, Stockholm, Sweden*

Abstract: Changes in the topology of communication networks, such as sudden appearance or disappearance of links or nodes, may signal malicious attacks or malfunctions. A topology change detector may thus be useful to trigger alarms or self-reconfiguration procedures. Here we present a novel approach that enjoys several desirable qualities such as fast convergence, intrinsically distributed computations, and scalability w.r.t. communication and computational requirements. We characterize the performance of this technique from analytical and practical points of view, providing theoretical results on its performance. We thus show how it is possible to tune and trade-off the accuracy of the change detection results with the communication requirements of the procedure.

Keywords: Networks topology inference, change detection, distributed computations, anonymous computations.

1. INTRODUCTION

The applicability of distributed systems to real frameworks, like in sensing, surveillance, monitoring and tracking (Akyildiz et al., 2002), is proportional to the amount of direct human intervention that these systems require for their operation. Implementing truly smart sensors and actuators networks requires thus opportune self-diagnosis tools, that lessen or ease maintenance tasks.

In this respect, the structure of the communication topology is of particular importance, since it represents one of the most affecting quantities of a distributed system (e.g., it limits how and how fast the information can flow among the agents (Li and Yang, 2006)). Consider then that in a distributed scenario it may not be convenient to rely on a communication topology that is defined during the design phase; it may indeed be that the network structure needs to be inferred after the deployment phase. There is, in fact, an active research community focusing on such *graph discovery* algorithms (Parlangeli and Notarstefano, 2012; Shames et al., 2012; Sahai et al., 2012). These algorithms can be used to build, on top of them, automated topology change detectors, so that agents of a network can discover changes in its connectivity properties and react by

triggering specialized problem-detection procedures, e.g., searches for coverage or routing holes (Ahmed et al., 2005), and, in cascade, strategies for the reconfiguration of the network itself (Wang, 2008; Chen et al., 2002).

Our aim here is to design a topology change detector that exploits a particular data aggregation scheme, namely max-consensus over random vectors, that is often used for size estimation purposes (Cichon et al., 2011; Baquero et al., 2012). With this choice, the network change detection algorithm benefits directly of several desirable properties, namely fast propagation of the information across the network, little memory / computational and communication requirements, scalability, robustness w.r.t. nodes and links failures, full parallelism (i.e., every agent eventually computes the same estimates), and not requiring a leader election step. Moreover, as it will be clear later, the proposed technique is intrinsically robust to churn since it continuously provides aggregated information to all nodes.

Literature review: the detection of topological changes can be based on comparing subsequent estimates of some properties of the communication graph. Ideally it is thus possible to convert each graph discovering algorithm into a topological change detector.

There is a vast literature on graph discovering, and different approaches deal with different types of constraints. A classical and intuitive strategy is to exchange *tables of IDs* and from them reconstruct the whole topology (Deb et al., 2004). These strategies can give reliable results but are neither scalable nor distributed.

^{*} The research leading to these results has received funding from the European Union Seventh Framework Programme [FP7/2007-2013] under grant agreement n°257462 HYCON2 Network of excellence, the European Institute of Technology (EIT) Information and Communication Technology (ICT) Labs, the Swedish Energy Agency, the Swedish Governmental Agency for Innovation Systems (VINNOVA), the Swedish Research Council and the Knut and Alice Wallenberg Foundation. Corresponding author: D. Varagnolo, damiano.varagnolo@1tu.se

A truly distributed approach is instead to rely on *random walk techniques* (Hall, 2010; Kurant et al., 2012; Massoulié et al., 2006), i.e., on passing a token through the network. This token collects information each time it visits an agent through two mechanisms: either it counts how many hops it took to return to the sender, or it counts how much time it is required for a given counter to become zero. Statistical properties of the return-time and time-to-vanish are then used to infer the network size.

A third scheme is to exploit *capture-recapture strategies*, i.e., disseminate a certain number of messages called “seeds”, propagate them through the network, query some pre-determined nodes whether they hold a seed or not and then infer the size of the network from the number of seeds in the set of queried nodes (Peng et al., 2009; Petrovic and Brown, 2009; Bawa et al., 2003).

We incidentally report also a class of algorithms that directly focus on detecting topological changes in Wireless Sensor Networks (WSNs), that are explicitly based on measuring geographical phenomena, e.g., wildfires (Farah et al., 2011; Fekete et al., 2004; Jiang and Worboys, 2009). Here, instead, we consider the more general case where environmental or similar measurements are not available to the change detector.

Statement of contributions: we derive a Generalized Likelihood Ratio (GLR)-based change detection algorithm on top of a particular and already existing size estimation procedure, employing the fastest distributed aggregation mechanism existing. The proposed innovation is thus: provide the full description of the procedure in all its implementation details; derive its statistical characterization and the trade-offs between the computational / communication complexities and the detection performance; corroborate the findings with numerical experiments; provide indications and recommendations for the usage in practical situations.

Organization of the manuscript: Section 2 collects assumptions and notation, recalls and summarizes basic definitions on the theory on hypotheses testing, and then formulates the change detection problem. Section 3 reports how to estimate cardinalities using max-consensus protocols, describes the change detection algorithm and motivates our choices. Section 4 describes the statistical properties of the change detector, analyses the existing trade-offs between communication complexities and accuracy of the results and shows the performance of our technique on simulated case studies. Section 5 finally collects some concluding remarks and future research directions. For ease of readability, all the proofs are collected in the appendix.

2. ASSUMPTIONS, NOTATION AND PROBLEM FORMULATION

In the following, we consider the simplified framework where the effects of packet losses and quantization issues can be neglected. Moreover, we assume the following communication protocol: *time is divided in epochs, indexed by $t = 0, 1, 2, \dots$. Every agent broadcasts its information exactly once per epoch. The order of the broadcasting operations is irrelevant, and can change in time. When an agent*

broadcasts its information, it broadcasts the information that it had at the beginning of the epoch.

Notice that the time index t does *not* measure a physical quantity (e.g., seconds), but rather the index of the various epochs, the latter defined as the time necessary to let all the agents communicate exactly once.

With reference to the information generation scheme discussed later we denote by $\{\mathcal{G}(t) := (\mathcal{V}(t), \mathcal{E}(t))\}$ the sequence of network graphs associated to the execution of Algorithm 1. $\mathcal{V}(t)$ represents the set of active agents at time t . $\mathcal{E}(t) \subseteq \mathcal{V}(t) \times \mathcal{V}(t)$ represents the set of communications among active agents at time t (as described in step 3 of Algorithm 1): $(i, j) \in \mathcal{E}(t)$ indicates that in period t agent i successfully broadcasts its information to j . For notational simplicity, we will often identify agents with the fictitious indexes i and j .

The set of k -steps neighbors of agent i at time t is defined for $k = 0$ as $\mathcal{V}_0^{(i)}(t) := \{i\}$ and, for $k \geq 1$, through the recursion

$$\mathcal{V}_k^{(i)}(t) := \bigcup_{(i,j) \in \mathcal{E}(t)} \mathcal{V}_{k-1}^{(j)}(t-1). \quad (1)$$

A pictorial representation of k -steps neighborhoods is given in Figure 1.

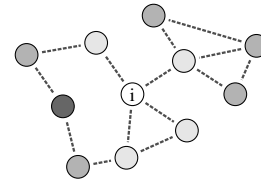


Fig. 1. Example of k -steps neighborhoods. Assuming a time-invariant network, it holds that $\mathcal{V}_1^{(i)}$ is given by \textcircled{i} and all the nodes $\textcircled{}$. $\mathcal{V}_2^{(i)}$, instead, comprises $\mathcal{V}_1^{(i)}$ and all the nodes \bullet .

In the following the cardinality of $\mathcal{V}_k^{(i)}(t)$ is denoted by $S_k^{(i)}(t)$.

2.1 Notation on hypothesis testing

Let bold fonts indicate vectors, plain italic fonts indicate scalars and capitalized plain italic fonts indicate matrices. We denote by either $p(x; \theta)$ or $p_\theta(x)$ the probability density of the random variable (r.v.) x parametrized by the scalar θ . $\ell(\theta; x)$ is the likelihood of θ given x . Consider two complementary hypotheses, \mathcal{H}_0 and \mathcal{H}_1 , on how a r.v. \mathbf{f} has been generated, of the form

$$\mathcal{H}_i = \{\mathbf{f} \sim p_\theta \text{ with } \theta \in \Theta_i\}, \quad i = 0, 1, \quad \Theta_1 = \Theta_0^c$$

where \cdot^c denotes complementation. Selecting \mathcal{H}_1 when \mathcal{H}_0 is true is said to be an *error of type I*. Conversely, accepting \mathcal{H}_0 when \mathcal{H}_1 is true is said to be an *error of type II*. Let the selection between \mathcal{H}_0 and \mathcal{H}_1 be performed through a deterministic function $g(\mathbf{f})$ with range $\{0, 1\}$. g thus partitions the space of the plausible outcomes for \mathbf{f} in two regions: the *acceptance region* $R := \{\mathbf{f} \text{ s.t. } g(\mathbf{f}) = 0\}$ and the *critical region* R^c . The function

$$\beta_g(\theta) := \mathbb{P}[\mathbf{f} \in R^c; \theta] \quad (2)$$

is called the *power function* of g . This map characterizes the statistical performance of g through

$$\alpha_0(g) := \sup_{\theta \in \Theta_0} \mathbb{P}[\mathbf{f} \in R^c ; \theta] , \quad (3)$$

called the *size* of g , namely the worst probability of errors of type I given all the possible instances of $\theta \in \Theta_0$. If $\theta \in \Theta_1$, instead, $\beta_g(\theta)$ is the probability of *not* committing errors of type II for that particular θ .

For details see, e.g., (Basseville and Nikiforov, 1993; Lehmann and Romano, 2005).

2.2 Problem formulation

Assume that the agent i and the neighborhood of interest k are given, and consider the finite time horizon $t - N, \dots, t$, with N fixed. We formulate the problem of detecting topological changes over this finite window of time (dropping here for notational simplicity the subscripts and superscripts), as the one of discerning between the two composite hypotheses

$$\begin{cases} \mathcal{H}_0 : S(t - N) = \dots = S(t - T) = \bar{S}, \\ \quad S(i) \geq \sigma \bar{S} \text{ for all } i \in \{t - T + 1, \dots, t\} \\ \mathcal{H}_1 : S(t - N) = \dots = S(t - T) = \bar{S}, \\ \quad \exists i \in \{t - T + 1, \dots, t\} \text{ s.t. } S(i) < \sigma \bar{S} \end{cases} \quad (4)$$

The change detection hypotheses are thus parametrized both in $T \in \{1, \dots, N\}$ and in $\sigma, \bar{S} \in \mathbb{R}_+$. The interpretation is the one of testing a trend of decrease for the k -steps neighborhood size $S(t)$, with the change amplitude dominated by the parameter $\sigma \in (0, 1)$. Specifically, \mathcal{H}_0 corresponds to a nominal operating condition, while \mathcal{H}_1 describes the disconnection of nodes in the given neighborhood.

The change detection hypotheses involve thus the post-change values and 3 parameters, the pre-change value \bar{S} , the jump amplitude σ and the time of change T .

3. A NEW ALGORITHM FOR THE DETECTION OF TOPOLOGICAL CHANGES

Intuitively, detecting changes in the size of k -steps neighborhoods can be done by continuously estimating it and inspecting the temporal evolution of the estimates. The aim of this section is to demonstrate how this is possible by cascading some statistical inference tools with a Generalized Likelihood Ratio (GLR) approach.

3.1 Continuous estimation of topological properties

We propose to continuously estimate a given neighborhood size by exploiting a probabilistic counting algorithm that is based on max consensus and works as follows. Let each agent i be endowed with a time-varying matrix $F^{(i)}(t) \in \mathbb{R}^{M \times D}$, where the dimensions M and D are fixed a-priori, and t denotes the epoch index. For notational brevity, we let $\mathbf{f}_k^{(i)}(t)$ denote the k -th column of $F^{(i)}(t)$, and $f_{m,k}^{(i)}(t)$ the element in the m -th row and k -th column of $F^{(i)}(t)$ ($k = 1, \dots, D$, $m = 1, \dots, M$). Let then agents exchange information as in Algorithm 1.

Since the communication protocol in Section 2 makes the agents broadcast the $F^{(i)}(t)$ they computed *at the beginning* of the loop, and not after the max-consensus

Algorithm 1 Information generation scheme

- 1: **for** $t = 1, 2, \dots$ **do**
 - 2: (*Information Update*) each agent i computes $F^{(i)}(t)$ by shifting the columns of $F^{(i)}(t - 1)$, in the sense that $\mathbf{f}_k^{(i)}(t) = \mathbf{f}_{k-1}^{(i)}(t - 1)$ for $k = 2, \dots, D$. $\mathbf{f}_1^{(i)}(t)$ is instead filled with M new i.i.d. random values $f_{m,1}^{(i)}(t) \sim \mathcal{U}[0, 1]$, $m = 1, \dots, M$;
 - 3: (*Communication*) every agent broadcasts $F^{(i)}(t)$ to its neighbors;
 - 4: (*Information Mixing*) every agent updates its $F^{(i)}(t)$ by means of the $F^{(j)}(t)$'s received from its neighbors. More specifically, if $\mathcal{N}_i(t)$ is the set of the neighbors j from which i received information, then
$$\mathbf{f}_{m,k}^{(i)}(t) \leftarrow \max_{j \in \mathcal{N}_i(t)} \left(\mathbf{f}_{m,k}^{(i)}(t), \left\{ \mathbf{f}_{m,k}^{(j)}(t) \right\} \right) \quad (5)$$
for $m = 1, \dots, M$, $k = 1, \dots, D$.
 - 5: **end for**
-

performed in Equation (5), $\mathbf{f}_k^{(i)}(t)$ aggregates information exactly from the agents in the k -steps neighborhood $\mathcal{V}_k^{(i)}(t)$. It follows from (Varagnolo et al., 2010) that the Maximum Likelihood (ML) estimator of $S_k^{(i)}(t)$ is given by

$$\begin{aligned} \widehat{S}_k^{(i)}(t) &:= \left(-\frac{1}{M} \sum_{m=1}^M \log \left(f_{m,k}^{(i)}(t) \right) \right)^{-1} \\ &= \left(\chi_k^{(i)}(t) \right)^{-1}, \end{aligned} \quad (6)$$

where $\chi_k^{(i)}(t) \sim \text{Gamma} \left(M, \left(M S_k^{(i)}(t) \right)^{-1} \right)^1$.

Moreover (consequence that will be useful later) if $t_1 : t_2$ stands for $t_1, t_1 + 1, \dots, t_2$, then, under the hypothesis that $S_k^{(i)}(t)$ remains constant at times $t \in \{t_1 : t_2\}$, its ML estimator is

$$\widehat{S}_k^{(i)}(t_1 : t_2) = \left(\frac{1}{t_2 - t_1 + 1} \sum_{t=t_1}^{t_2} \chi_k^{(i)}(t) \right)^{-1}.$$

3.2 The change detection algorithm

Assume w.l.o.g. to focus on a particular agent i and neighborhood k , so that for notational brevity we can drop all the superscripts (i) and subscripts k , and indicate $\mathbf{f}_k^{(i)}(t)$ with $\mathbf{f}(t)$, $\chi_k^{(i)}(t)$ with $\chi(t)$, $\mathcal{V}_k^{(i)}(t)$ with $\mathcal{V}(t)$ and $S_k^{(i)}(t)$ with $S(t)$.

Consider again the hypotheses \mathcal{H}_0 and \mathcal{H}_1 defined in (4) with σ and N fixed. The previous estimation algorithms lead thus to the following Algorithm 2, to be executed for every t in Algorithm 1. The change detection algorithm works as follows: start by computing, for all the plausible change times, the ML estimates of the pre-change and under the post-change values under no constraints and under hypothesis \mathcal{H}_0 (Equations (7), (8) and (9) respectively). Then compute all the possible GLRs in (10), from which it is possible to estimate the most likely change time in (11) and thus decide between the hypotheses in (12).

¹ We denote by Gamma (a, b) the distribution of a gamma r.v. with shape a and rate b .

Algorithm 2 Neighborhood size change detection

(cycle on all the plausible change times)

1: **for** $\mathcal{T} = 1, \dots, N - 1$ **do**

2: (estimation of the pre-change value)

$$\bar{S}(\mathcal{T}) = \left(\frac{1}{N - \mathcal{T} + 1} \sum_{\tau=t-N}^{t-\mathcal{T}} \chi(\tau) \right)^{-1} \quad (7)$$

3: **for** $\tau \in \{t - \mathcal{T} + 1, \dots, t\}$ **do**

4: (estimation of the post-change values under no hypotheses and under \mathcal{H}_0)

$$\hat{S}(\tau) = \chi(\tau)^{-1} \quad (8)$$

$$\hat{S}_0(\tau) = \begin{cases} \hat{S}(\tau) & \text{if } \hat{S}(\tau) \geq \sigma \bar{S}(\mathcal{T}) \\ \sigma \bar{S}(\mathcal{T}) & \text{otherwise} \end{cases} \quad (9)$$

5: **end for**

6: (computation of the log-GLR)

$$\Lambda(\mathcal{T}) = M \sum_{\tau=t-\mathcal{T}+1}^t \log \left(\frac{\hat{S}_0(\tau)}{\hat{S}(\tau)} \right) - (\hat{S}_0(\tau) - \hat{S}(\tau)) \chi(\tau) \quad (10)$$

7: **end for**

8: (computation of the optimal change time)

$$T = \arg \min_{\mathcal{T} \in \{1, \dots, N-1\}} \Lambda(\mathcal{T}) \quad (11)$$

9: (decision between \mathcal{H}_0 and \mathcal{H}_1)

$$g(\mathbf{f}(t - N : t)) = \begin{cases} 0 & \text{if } \Lambda(T) \geq \lambda_T \\ 1 & \text{otherwise.} \end{cases} \quad (12)$$

Notice that the thresholds λ_T , $T \in \{1, \dots, N - 1\}$, implicitly define the power of the hypothesis test, and should be chosen according to the desired size α_0 . How to compute the test thresholds is the next subject of our discussion.

4. CHARACTERIZATION OF ALGORITHM 2

Here we derive the statistical properties of the change detection rule (12). We start by stating formally its theoretical performance as a function of the thresholds λ_T and then, to confirm the theoretical findings, we apply the change detector to two simulated case studies.

4.1 Statistical properties

In the following we assume that \bar{S} and T are fixed deterministic parameters. We want to derive the statistical characterization of the test power function $\beta_g(\theta)$ defined in (2). To this aim, we notice that g depends on the test thresholds λ_T in (12), and that these are in turn function of the desired test size α_0 , defined in (3).

Although the map from α_0 to the thresholds λ_T has no closed-form expressions, it can nonetheless be evaluated through an opportune numerical procedure:

Proposition 1. Let α_0 be the desired size of the test defined in Algorithm 2. Then λ_T in (12) can be computed as a function of α_0 by means of Algorithm 3.

The dependence of the test thresholds λ_T on the change times T is highlighted in step 5 of Algorithm 3. We also

Algorithm 3 Computation of the thresholds λ_T

1: compute

$$q_1 = 1 - \frac{\Gamma(M, M)}{\Gamma(M)} \quad (13)$$

where $\Gamma(a, b)$ is the upper incomplete Gamma function and $\Gamma(a)$ is the Gamma function

2: set $p_1(a)$ to be the probability density of a Gamma random variable with shape M and scale M^{-1} (i.e., $\text{Gamma}(M, M^{-1})$) for $a \in (0, 1]$

3: compute $p_2(a)$ as

$$p_2(a) = p_1(-W(-e^{-a-1})) W'(-e^{-a-1}) e^{a-1}$$

4: set $p_\nu(a)$ as the mixed probability density and mass function of the novel r.v.

$$\nu = \begin{cases} 1 & \text{with mass } q_1 \\ a \in (0, 1) & \text{with density } p_2(a) / q_1 \end{cases}$$

5: compute the mixed probability density and mass function of the novel r.v. ω as

$$p_\omega(\cdot) = \overbrace{p_\nu(\cdot) * \dots * p_\nu(\cdot)}^{T \text{ times}},$$

where $*$ denotes convolution (c.f. von Lanzener and Lundberg (1974))

6: compute the quantile function of ω , say $F_\omega^{-1}(\cdot)$

7: compute λ_T as $\lambda_T = F_\omega^{-1}(\alpha_0)$

notice that these thresholds do not depend on the actual value of $\sigma \bar{S}$. This mirrors the fact that the test checks for *relative* variations in amplitude of the network rather than for *absolute* variations. I.e., the test checks whether the total size diminished by, e.g., one fifth rather than 20 units.

We now proceed to study the dependency of the power function $\beta_g(\theta)$ on M , i.e., the trade-offs between the probability of errors of type I and II (the power function) and the communication bandwidth requirements (the number of random numbers to be generated M , that is proportional to the packets size). The aim of this analysis is to provide directions on how to choose M , i.e., the number of r.v.'s that should be generated for each epoch in Algorithm 1, in order to achieve the desired error rates.

Let then α_0 (the worst probability of type I errors) and σ (the amplitude of the variations considered interesting) be given. Due to the multidimensionality of the parameters space it is not possible to analytically describe the whole β_g in (2) as a function of M and the realization $S(t - T + 1), \dots, S(t)$. We thus characterize the following reduced power function, where for notational simplicity we omit the dependencies on $\sigma, \bar{S}, \alpha_0$ and T :

$$\beta_g^r(\kappa, M) := \beta_g([\kappa \sigma \bar{S}, \dots, \kappa \sigma \bar{S}]; M) = \mathbb{P}[\mathbf{f}(t - T + 1 : t) \in R^c; \kappa \sigma \bar{S}]. \quad (14)$$

In (14), κ describes the amplitude of the jump. Although $\beta_g^r(\kappa, M)$ cannot exhaustively represent the performance of the test, nonetheless it gives indications about the considered trade-offs.

The computation of β_g^r resembles the computation of the thresholds λ_T and, as before, does not rely on closed-form expressions but rather on a opportune numerical procedure:

Proposition 2. The reduced power function $\beta_g^r(\kappa, M)$ defined in (14) can be evaluated by means of Algorithm 4.

Algorithm 4 Computation of $\beta_g^r(\kappa, M)$ (σ, α_0 and T given)

1: compute

$$q_1 = 1 - \frac{\Gamma(M, \kappa M)}{\Gamma(M)}$$

as in step 1 of Algorithm 3

2: set $p_1(a)$ to be the probability density of a Gamma random variable with shape M and scale $(\kappa M)^{-1}$ (i.e., Gamma $(M, (\kappa M)^{-1})$) for $a \in (0, 1]$

3: perform steps 2 - 3 - 4 - 5 - 6 of Algorithm 3 (in step 6 compute the cumulative distribution $F_\omega(\cdot)$ rather than the quantile function $F_\omega^{-1}(\cdot)$)

4: compute

$$\beta_g^r(\kappa, M) = F_\omega(\lambda_T)$$

By construction, β_g^r depends only on κ . Its behavior is shown in Figure 2.

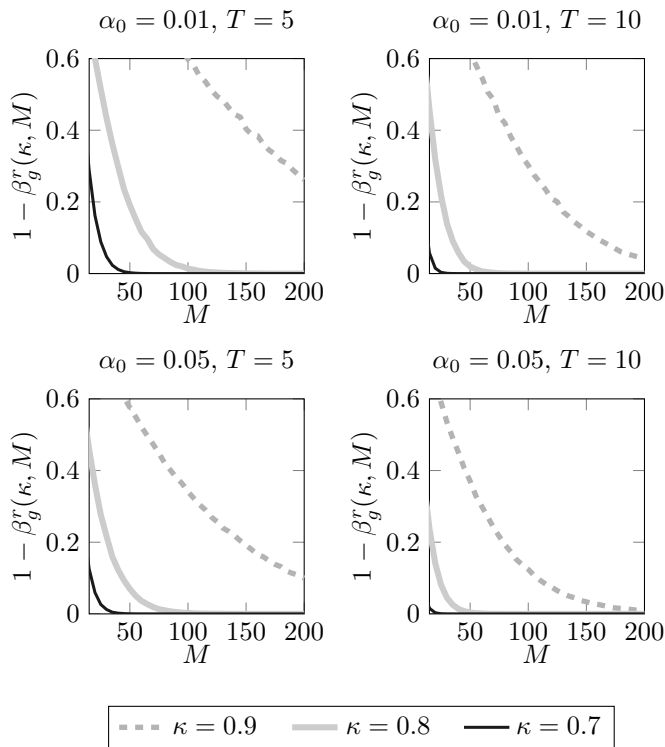


Fig. 2. Dependence of $\beta_g^r(\kappa, M)$ defined in (14) on κ and M for various cases of α_0, T . Given $\kappa < 1$, $1 - \beta_g^r(\kappa, M)$ describes the probability of committing type II errors (i.e., false negatives).

Notice that, given (4), $\kappa < 1$ implies to be considering \mathcal{H}_1 , so that β_g^r returns the probability of not committing type II errors (or, equivalently, $1 - \beta_g^r$ is the probability of committing false negatives). As expected, the probability of committing type II errors is inversely proportional with M, T and the size α_0 , and directly proportional with κ . We remark that the quick decay of $1 - \beta_g^r$ with M is

independent of the actual sizes and number of links in the various neighborhoods of the network.

4.2 Performance in synthetic networks

We now analyze the performance of the algorithm in simulated networks. We notice that, from an algorithmic point of view, in step 1 of Algorithm 2 and in (11) we change the condition $\mathcal{T} \in \{1, \dots, N - 1\}$ with

$$\mathcal{T} \in \{1, \dots, N - \bar{N}\}, \quad \bar{N} > 1 \quad (15)$$

so that the estimation of the pre-change value is less sensitive to outliers (i.e., we avoid to compute those estimates (7) for which there is little information available).

We focus our attention on the 2 following scenarios:

Clustered networks The first example, see Figure 3a, is a network formed by three cliques, each of six nodes, connected by hubs. Initially all the nodes are active and run Algorithm 1 and Algorithm 2 in parallel. Then, at time t^* , an hub fails, drastically changing the topology of the network (e.g., the pre-change diameter is 3, the post-change 5).

Figure 3a shows the evolution of the alarms generated by the various nodes in a typical realization of the change-detection algorithm. White nodes are nodes that have not detected changes. Colored nodes instead are nodes that detected some changes: more precisely, the number inside the circle indicates the smallest neighborhood for which changes have been detected. E.g., in Figure 3a, at the end of epoch $t^* + 1$, every node in the whole upper clique has detected a change in its 1-step neighborhood. Although not shown in the figure, we notice that the nodes remain in alarmed states for approximately N epochs, i.e., the time period for which pre-change information is still available in the node.

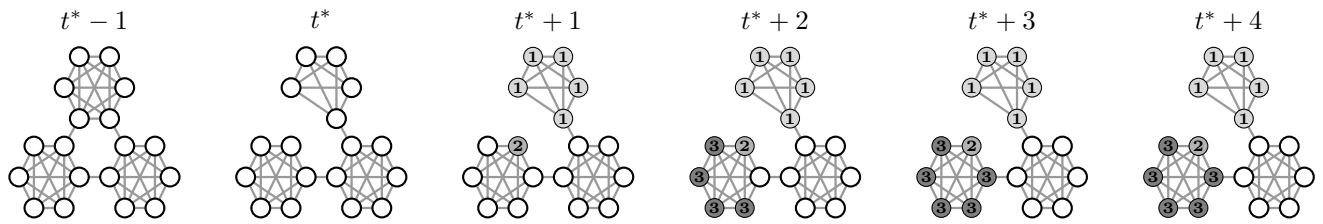
Grid networks The second example, see Figure 3b, is a 2 dimensional grid. As before, initially all the nodes are active and run Algorithm 1 and Algorithm 2 in parallel. Then, at time t^* , some communication links fail, letting the communication graph resemble now a comb.

As time passes, the nodes in the teetles of the comb (for which the sizes of the various k -steps neighborhoods have drastically changed) start recognizing the change. As expected, instead, the nodes in the handle of the comb do not detect the topological modification.

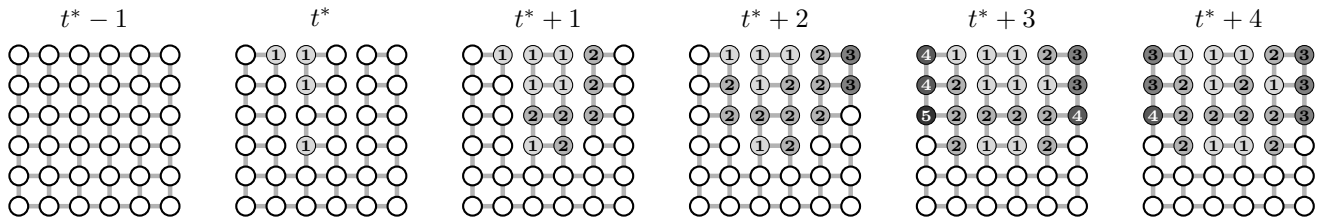
5. CONCLUSIONS

We presented a network topology change detection technique that relies on synchronous communications assumptions and max consensus protocols. More specifically, the algorithm exploits statistical size estimation techniques that are intrinsically and naturally distributed.

The method requires the user to set just two nominal parameters, i.e., the amplitude of the changes to be detected and the worst probability of errors of type I accepted. All the other parameters are estimated either off-line or on-line, thus making the strategy easy to be used in real applications. Moreover the algorithm exploits max-consensus



(a) change detection in a simulated clustered network where at time t^* a hub fails. Here $\alpha_0 = 0.01$, $M = 100$, $\sigma = 0.9$, $N = 20$ and $\bar{N} = 5$.



(b) example in a simulated grid network where at time t^* a set of communication links fail. Here $\alpha_0 = 0.01$, $M = 100$, $\sigma = 0.9$, $N = 20$ and $\bar{N} = 5$.

Fig. 3. Experiments on two simulated networks. White nodes indicate agents that have not detected changes. Colored nodes instead indicate agents detecting a change in at least one of their k -steps neighborhoods (the number indicates the smallest neighborhood for which the agent detected a change).

communication protocols, the fastest information aggregation scheme, and thus addresses situations where the speed of detection is crucial.

The strategy has been shown to be effective in detecting changes in communication networks, and this suggests to continue paying efforts to improve it. More specifically, we suggest to remove synchronous assumptions and to extend it to detect changes of other aggregated quantities, like networks diameters, radii and eccentricities. Moreover we believe that it will be beneficial to implement it in real testbeds, e.g., in transportation systems, and to combine this change detector with opportune network reconfiguration strategies.

REFERENCES

- Nadeem Ahmed, Salil S. Kanhere, and Sanjay Jha. The holes problem in wireless sensor networks. *ACM SIGMOBILE Mobile Computing and Communications Review*, 9(2):1 – 14, April 2005.
- I.F. F Akyildiz, Weilian Su, Y. Sankarasubramaniam, and E. Cayirci. A survey on sensor networks. *IEEE Communications Magazine*, 40(8):102–114, August 2002.
- Carlos Baquero, Paulo Sergio Sérgio Almeida, Raquel Menezes, and Paulo Jesus. Extrema Propagation: Fast Distributed Estimation of Sums and Network Sizes. *IEEE Transactions on Parallel and Distributed Systems*, 23(4):668 – 675, April 2012.
- Michele Basseville and I. V. Igor V Nikiforov. *Detection of Abrupt Changes: theory and application*. Prentice-Hall, April 1993.
- Mayank Bawa, Hector Garcia-molina, Aristides Gionis, and Rajeev Motwani. Estimating aggregates on a peer-to-peer network. Technical report, Stanford InfoLab, April 2003.
- Benjie Chen, Kyle Jamieson, Hari Balakrishnan, and Robert Morris. Span: An Energy-Efficient Coordination Algorithm for Topology Maintenance in Ad Hoc Wireless Networks. *Wireless Networks*, 8(5):481–494, September 2002.
- Jacek Cichon, Jakub Lemiesz, and Marcin Zawada. On Cardinality Estimation Protocols for Wireless Sensor Networks. *Ad-hoc, mobile, and wireless networks*, 6811: 322–331, 2011.
- Budhaditya Deb, Sudeept Bhatnagar, and Badri Nath. STREAM: Sensor Topology Retrieval at Multiple Resolutions. *Telecommunication Systems*, 26(2-4):285–320, June 2004.
- Christopher Farah, Frederick Schwaner, Ali Abedi, and Michael Worboys. Distributed Homology Algorithm to Detect Topological Events via Wireless Sensor Networks. *Wireless Sensor Systems, IET*, 1(3):151 – 160, 2011.
- Sandor P. Fekete, Alexander Kroeller, Dennis Pfisterer, Stefan Fischer, and Carsten Buschmann. Neighborhood-Based Topology Recognition in Sensor Networks. In *Workshop on Algorithms Aspects of Sensor Networks*, page 14, May 2004.
- Cyrus P Hall. *Peer-to-Peer Algorithms for Sampling Generic Topologies*. PhD thesis, Università della Svizzera Italiana, 2010.
- Jixiang Jiang and Michael Worboys. Preliminaries for Topological Change Detection Using Sensor Networks. In Niki Trigoni, Andrew Markham, and Sarfraz Nawaz, editors, *Geosensor Networks*, volume 5659 of *Lecture Notes in Computer Science*, pages 112–121. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.
- Maciej Kurant, Carter T. Butts, and Athina Markopoulou. Graph Size Estimation. *arXiv*, October 2012.
- E.L. Lehmann and Joseph P. Romano. *Testing Statistical Hypotheses*. Springer, 2005.
- Mo Li and Baijian Yang. A Survey on Topology issues in Wireless Sensor Network. In *Proceedings of the International Conference on Wireless Networks*, 2006.
- Laurent Massoulié, Erwan Le Merrer, Anne-Marie Ker-marrec, and Ayalvadi Ganesh. Peer counting and sam-

pling in overlay networks: random walk methods. In *Proceedings of the twenty-fifth annual ACM symposium on Principles of distributed computing*, pages 123–132, 2006.

Gianfranco Parlangeli and Giuseppe Notarstefano. On the Reachability and Observability of Path and Cycle Graphs. *IEEE Transactions on Automatic Control*, 57(3):743–748, March 2012.

Shao-Liang Peng, Shan-Shan Li, Xiang-Ke Liao, Yu-Xing Peng, and Nong Xiao. Estimation of a Population Size in Large-Scale Wireless Sensor Networks. *Journal of Computer Science and Technology*, 24(5):987–997, September 2009.

Sanja Petrovic and Patrick Brown. A new statistical approach to estimate global file populations in the eDonkey P2P file sharing system. In *21st International Teletraffic Congress*, September 2009.

Tuhin Sahai, Alberto Speranzon, and Andrzej Banaszuk. Hearing the clusters of a graph: A distributed algorithm. *Automatica*, 48(1):15–24, January 2012.

Iman Shames, Themistoklis Charalambous, Christoforos N. Hadjicostis, and Mikael Johansson. Distributed Network Size Estimation and Average Degree Estimation and Control in Networks Isomorphic to Directed Graphs. In *Allerton Conference on Communication Control and Computing*, 2012.

Damiano Varagnolo, Gianluigi Pillonetto, and Luca Schenato. Distributed statistical estimation of the number of nodes in Sensor Networks. In *IEEE Conference on Decision and Control*, pages 1498–1503, Atlanta, USA, December 2010.

Christoph Haehling von Lanzener and William N. Lundberg. The n-Fold Convolution of a Mixed Density and Mass Function. *The Journal of the International Actuarial Association*, 8(1):91–103, September 1974.

Yu Wang. Topology Control for Wireless Sensor Networks. *Wireless Sensor Networks and Applications*, 2:113–147, 2008.

APPENDIX

The proof of 1 requires the following lemmas, provided without proof due to space constraints.

Lemma 3. Consider g in (12) as a function of the T -dimensional vector $[S(t-T+1), \dots, S(t)]$, with each scalar component in \mathbb{R}_+ . Then for any $\tau \in \{t-T+1, \dots, t\}$ and fixed $\bar{S}(t-T+1), \dots, \bar{S}(\tau-1), \bar{S}(\tau+1), \dots, \bar{S}(t)$ the power of

$$g(\bar{S}(t-T+1), \dots, \bar{S}(\tau-1), S(\tau), \bar{S}(\tau+1), \dots, \bar{S}(t))$$

is strictly monotone decreasing with $S(\tau)$.

Lemma 4. The size of test (12) can be computed by evaluating the power β_g at the boundary $\sigma\bar{S}$, i.e.,

$$\alpha_0 = \beta_g(S(t-N : t-T) = \bar{S}, S(t-T+1 : t) = \sigma\bar{S}). \quad (16)$$

Lemma 5. Let $y = f(x) = 1 - x + \log x$. Then $f^{-1}(y) = -W(-e^{y-1})$ where $W(\cdot)$ is the Lambert W -function, i.e., the family of functions implicitly defined by $z = W(z)e^{W(z)}$.

Proof. (of Proposition 1) The aim of the proof is to find a numerical procedure that allows the user to compute λ_T once α_0 has been fixed. Let

$$\begin{aligned} \eta(\tau) &:= \chi(\tau)\widehat{S}_0(\tau), \\ \omega(\tau) &:= M\left(\log(\eta(\tau)) - \eta(\tau) + 1\right), \\ \omega &:= \sum_{\tau=t-T+1}^t \omega(\tau). \end{aligned} \quad (17)$$

Notice that all these quantities are mixed r.v.'s: $\eta(\tau)$ is discrete in 1, so that $\omega(\tau)$ is discrete in 0. With these definitions it follows that once the desired size α_0 has been fixed, λ_T in (12) is implicitly defined by $\mathbb{P}[\omega < \lambda_T; \sigma\bar{S}] = \alpha_0$. Since the various $f(\tau)$'s are i.i.d., also the $\omega(\tau)$'s are i.i.d. Moreover, the mixed density and mass function of $\sum_{\tau=t-T+1}^t \omega(\tau)$ can be computed as the T -fold convolution of $p(\omega(\tau))$, see (von Lanzener and Lundberg, 1974).

Now notice that the continuous part of each $\omega(\tau)$ in differentiable as a function of $\eta(\tau)$ in \mathbb{R}^+ and that also its inverse is differentiable (by Lemma 5). Since the distribution of $\eta(\tau)$ is known, that of $\omega(\tau)$ can be computed in the usual manner as the sum of its mass and density parts. Specifically for the discrete part of $\eta(\tau)$ one has

$$q_1 := \mathbb{P}[\eta(\tau) = 1; \sigma\bar{S}] = \mathbb{P}[\widehat{S}(\tau) \geq \sigma\bar{S}; \sigma\bar{S}],$$

where (Varagnolo et al., 2010) $\widehat{S}(\tau)$ is distributed as Inv-Gamma $(M, M\sigma\bar{S})$. Since the cumulative distribution function of a r.v. $x \sim \text{Inv-Gamma}(a, b)$ is given by

$$\mathbb{P}[x \leq c] = \frac{\Gamma(a, b/c)}{\Gamma(a)},$$

where the numerator is the upper incomplete gamma function and the denominator is the gamma function, it follows

$$q_1 = 1 - \frac{\Gamma(M, M)}{\Gamma(M)}. \quad (18)$$

As for its continuous part, since $\eta(\tau) = \sigma\bar{S}\widehat{S}(\tau)^{-1}$ and $\widehat{S}(\tau)^{-1} \sim \text{Gamma}(M, (M\sigma\bar{S})^{-1})$, it follows that $\eta(\tau)|\eta(\tau) \in (0, 1) \sim \text{Gamma}(M, M^{-1})$. The unconditioned $\eta(\tau)$ has thus its density in $(0, 1)$ equal to $\text{Gamma}(M, M^{-1})/q_1$, with the normalization factor q_1 accounting for the discrete component.

It is thus possible to compute the mixed density and mass of $\eta(\tau)$ and then compute back all the previous quantities, as proposed in Algorithm 3.

Proof. (of Proposition 2) This proof shares the same architecture of the one of Proposition 1. The unique difference is in the mixed density and mass function of the r.v. $\eta(\tau)$.

Given κ , its discrete part can be computed considering that $\widehat{S}(\tau) \sim \text{Inv-Gamma}(M, M\kappa\sigma\bar{S})$, that $q_1 = \mathbb{P}[\widehat{S}(\tau) \geq \sigma\bar{S}; \kappa\sigma\bar{S}]$, so that $q_1 = 1 - \frac{\Gamma(M, \kappa M)}{\Gamma(M)}$.

Its continuous part instead can be computed considering that $\eta(\tau) = \sigma\bar{S}\widehat{S}(\tau)^{-1}$, $\widehat{S}(\tau)^{-1} \sim \text{Gamma}(M, (M\kappa\sigma\bar{S})^{-1})$ so that $\eta(\tau)|\eta(\tau) \in (0, 1) \sim \text{Gamma}(M, (\kappa M)^{-1})$. The unconditioned $\eta(\tau)$ has thus its density in $(0, 1)$ equal to $\text{Gamma}(M, (\kappa M)^{-1})/q_1$, with the normalization factor q_1 again accounting for the discrete component.