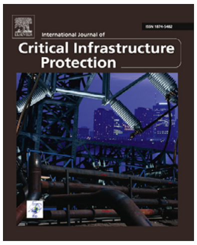
Available online at www.sciencedirect.com
ScienceDirect
journal homepage: www.elsevier.com/locate/IJCIP



Distributed monitoring for the prevention of cascading failures in operational power grids

Martijn Warnier[a,*], *Stefan Dulman*[b], *Yakup Koç*[a,c], *Eric Pauwels*[b]

[a] *Systems Engineering, Faculty of Technology, Policy and Management, Delft University of Technology, Jaffalaan 5, Delft 2628 BX, The Netherlands*
[b] *Intelligent Systems Group, Centrum Wiskunde and Informatica (CWI), Science Park 123, Amsterdam 1098 XG, The Netherlands*
[c] *Risk and Information Management, Stedin, Blaak 8, Rotterdam 3011 TA, The Netherlands*

ARTICLE INFO

Article history:
Received 7 April 2016
Revised 2 November 2016
Accepted 8 February 2017
Available online xxx

Keywords:
Power Grids
Cascading Failures
Robustness
Real-Time Monitoring
Distributed Computation

ABSTRACT

Electrical power grids are vulnerable to cascading failures that can lead to large blackouts. The detection and prevention of cascading failures in power grids are important problems. Currently, grid operators mainly monitor the states (loading levels) of individual components in a power grid. The complex architecture of a power grid, with its many interdependencies, makes it difficult to aggregate the data provided by local components in a meaningful and timely manner. Indeed, monitoring the resilience of an operational power grid to cascading failures is a major challenge.

This paper attempts to address this challenge. It presents a robustness metric based on the topology and operative state of a power grid to quantify the robustness of the grid. Also, it presents a distributed computation method with self-stabilizing properties that can be used for near real-time monitoring of grid robustness. The research thus provides insights into the resilience of a dynamic operational power grid to cascading failures during real-time in a manner that is both scalable and robust. Computations are pushed to the power grid network, making the results available at each node and enabling automated distributed control mechanisms to be implemented.

© 2017 Elsevier B.V. All rights reserved.

1. Introduction

Power grids are major critical infrastructure assets—all kinds of basic, government and private services depend on the continuous and reliable delivery of electricity. Power grid outages can have significant societal impacts in terms of human safety and economic losses.

The large-scale introduction of renewable energy sources and the current (centralized) architecture of the power grid increase the likelihood of power outages. Encouraged by government subsidies and the trend to become more "green," consumers are becoming producers of electricity by installing solar panels and wind mills [29]. Part of this produced power will be used locally, but excess power will be fed into the power grid. This contributes to grid instability [47] because it is difficult to predict and, thus, balance electricity production when power is supplied by a large number of small producers spread over a large geographical region, instead of a few large producers.

Unfortunately, the current power grid architecture does not support the large-scale introduction of small producers [1]. This significantly increases the possibility of a major power grid failure—initial local disruptions spread to the rest of a grid, evolving into a system-wide outage [10]. An initial failure can be caused by an external event such as a storm

[*] *Corresponding author.*
E-mail address: m.e.warnier@tudelft.nl (M. Warnier).

http://dx.doi.org/10.1016/j.ijcip.2017.03.003
1874-5482/© 2017 Elsevier B.V. All rights reserved.

and the failure effects can spread to the rest of the network in different ways, including voltage and frequency instabilities, hidden failures of protection systems, software and operator errors, and line overloads.

For example, a large-scale outage can be initiated by an overloaded line that is “tripped” by a circuit breaker. At this point, electricity can no longer flow through the line and the power flows to other lines. This can overload some of the lines, causing them to be tripped as well. As this process repeats over and over again, more and more lines are shut down, leading to a cascading failure of the power grid [13,39]. This paper focuses on cascading effects created by line overloads and on preventing cascading failures.

In order to detect (and ultimately prevent) cascading failures, it is necessary to monitor and alter the current state (power load distribution) of a power grid. The emerging smart grid is designed to do exactly this—it leverages a communications overlay that connects sensors and actuators. In effect, a smart grid is a large-scale distributed system that monitors line loads and accordingly changing the network state by tripping and untripping lines.

This paper deals with a smart grid environment. It focuses on two principal research questions. The first question is: What should be monitored? In other words, is there a metric that can predict cascading failures? The second question is: How should the grid be monitored? In other words, how should aggregation be performed and what is the appropriate temporal resolution for the monitoring? The extension of the resulting passive monitoring scheme to an active scheme that automatically alters the state of a power grid to prevent cascading failures is a topic for future research.

The main contribution of the paper is a new distributed monitoring approach that can be used to monitor the robustness of a power grid to cascading failures. The monitoring approach is based on the distributed computation of the robustness metric introduced in [24,25]. The contributions also include an extension of distributed gossip algorithms [9] with a self-stabilization mechanism to account for network dynamics. The resulting framework enables distributed aggregates to be computed in a rapid and reliable manner; this is at the heart of the proposed power grid monitoring approach.

The principal technical result is that it is possible to compute a complex robustness metric using simple, albeit robust, distributed primitives in a manner that makes the results readily available to every node in a power grid. The result is important because it enables the measurement infrastructure to be used in real-time to implement distributed control mechanisms for a power grid. The convergence time scales very well (logarithmic order) with respect to network size. The precision of the computations can be set by adjusting the message sizes and is independent of network parameters such as the number of nodes and network diameter.

2. Robustness metric

Several topological metrics have been proposed to express the vulnerability of a power grid to cascading failures. Examples include the average shortest path length, betweenness centrality [15] and gap metric [14]. These metrics can be used to determine the most critical nodes in a power grid.

However, in addition to its topological characteristics, a power grid has a physical aspect. In particular, electrical current in a power grid behaves according to Kirchhoff’s laws [5]. Therefore, a metric that is used to quantify the robustness of an operational power grid to cascading failures should consider its topological and physical characteristics.

The metric for robustness to cascading failures R_{CF} [24,25] does exactly this. It is, therefore, the starting point for the distributed power grid monitoring algorithm proposed in this paper. The robustness metric R_{CF} assesses the robustness of a power grid to cascading failures caused by line overloads. The metric relies on two main concepts: (i) electrical node robustness; and (ii) electrical node significance. Higher values of the R_{CF} metric indicate greater robustness, i.e., a power grid that is able to resist cascading failures to a larger extent. The remainder of this section summarizes previous work on robustness metrics. Interested readers are referred to [24,25] for additional details about robustness metrics.

2.1. Electrical node robustness

The electrical node robustness quantifies the ability of a bus (i.e., a node in a graph representation of a power grid) to resist cascading line overload failures by incorporating flow dynamics and network topology. Three key factors are used to calculate this robustness value for a node: (i) homogeneity of the load distribution on out-going branches (i.e., links in a graph representation of a power grid); (ii) loading level of the out-going links; and (iii) out-degree of the node.

Entropy is used to capture the first and third factors described above. The entropy of a load distribution at a node increases as flows over lines are distributed more homogeneously and the node out-degree increases. The entropy of a given load distribution at a node i is computed as:

$$H_i = -\sum_{j=1}^{d} p_{ij} \log p_{ij} \tag{1}$$

where d is the out-degree of the corresponding node and p_{ij} is the normalized flow value on the out-going link l_{ij}. The normalized flow value p_{ij} is computed as:

$$p_{ij} = \frac{f_{ij}}{\sum_{j=1}^{d} f_{ij}} \tag{2}$$

where f_{ij} is the flow value in line l_{ij}.

The effect of the loading level of the power grid is expressed using the tolerance parameter α (see [34]). The tolerance level α_{ij} of line l_{ij} is the ratio of the rated limit to the load of line l_{ij}. The parameter α is commonly used to compensate for the lack of data on the rated limits of components in test systems [44]. The analysis methods work such that, whenever the rated limits of grid components are known, the rated limits replace the α values.

Eqs. (1) and (2) are combined with the tolerance parameter α to capture the impact of the loading level on robustness. The resulting electrical robustness $R_{n,\,i}$ of a node i, which considers both the flow dynamics and the topological effects on network robustness, is given by:

$$R_{n,i} = -\sum_{j=1}^{d} \alpha_{ij} p_{ij} \log p_{ij} \tag{3}$$

The minus sign in Eq. (3) compensates for the negative electrical node robustness value that occurs when taking the logarithm of the normalized flow value. Note that only the out-degrees of nodes are considered in the formalization of electrical node robustness. The in-degree relates to the total amount of power flow to which a node can be exposed. In contrast, the out-degree of the same node relates to its ability transfer the power to the remainder of the network that has a relatively larger rest capacity to accommodate the excess power flow. Therefore, only the out-degree of a node is used to compute the electrical node robustness.

2.2. *Electrical node significance*

All the nodes in a power grid do not have the same influence on the occurrence of cascading failures. Some nodes distribute a relatively large amount of the power in the network whereas other nodes only distribute a small amount of power. When a node (or line to a node) that distributes a relatively large amount of power fails, the result is more likely to lead to a cascading failure, ultimately resulting in a large grid blackout. In contrast, if a node that distributes only a small amount of power fails, then the resulting redistribution of power can usually be accommodated by other parts of the network. Thus, node failures have different impacts on the robustness to cascading failures and the impacts depend on the amount of power distributed by the corresponding nodes.

The impact of a node i is expressed by its electrical node significance δ_i:

$$\delta_i = \frac{P_i}{\sum_{j=1}^{N} P_j} \tag{4}$$

where P_i is the total power distributed by node i and N is the number of nodes in the network.

Electrical node significance is a centrality measure that can be used to rank the relative importance (i.e., criticality) of nodes in a power grid with respect to cascading failures. Failures of nodes with higher δ values typically result in larger cascading failures.

2.3. *Network robustness metric*

The network robustness metric R_{CF} [24,25] is obtained by combining the node robustness and node significance:

$$R_{CF} = \sum_{i=1}^{N} R_{n,i}\delta_i \tag{5}$$

The network robustness metric can be used as an indicator of power grid robustness. This is accomplished by computing the robustness metric for a normally-operating power grid, which yields a value v. This value v is used as a base case. During normal operations, the robustness metric value changes somewhat because different nodes demand different amounts of electricity over time, leading to different loading levels in the network. However, a larger change in the robustness metric—a drop, in particular—indicates that a cascading failure is more likely and that power grid operators may need to take evasive actions (e.g., by adding reserve capacity to the grid or shifting power demand). Note that, in the general case, it is complicated to determine a good safety margin or the value of the robustness metric that corresponds to the exact tipping point (i.e., point where a small failure leads to a massive blackout). Ultimately, this needs to be determined by grid operators; in particular, they must identify what they consider to be acceptable and the appropriate safety margin.

This research determines this point experimentally by simulating a specific power grid (IEEE 118 Power System [12]) using the MATCASC tool [23]. However, this point needs to be determined experimentally for other grids as well. Interested readers are referred to [26,27] for a general and structured investigation of this topic.

3. Decentralized aggregation

Computing the robustness metric introduced in the previous section in a centralized manner raises a number of challenges when applied to large power grids that cover states, provinces or countries. Scalability, single-point-of-failure, real-time results dissemination, fault tolerance and maintenance of dedicated hardware are some of the requirements that suggest a decentralized approach over a centralized solution.

The problem of interest is modeled as a geometric random graph (mesh network) in which nodes mainly communicate with their immediate neighbors. The communications model assumes that time is discrete. During a time step, each node picks and communicates with a random neighbor. Major updates in the network occur just once in a while; for example, in the scenario of interest, new measurement data is made available every 15 min. The time rounds concept is employed and nodes are required to update their local data at the beginning of a round. The bootstrap problem and round-based time models have received coverage in the literature [6,19,35]. Since the constraints in the application scenario are very loose, an algorithm like the one presented by Werner-Allen et al. [41] may be used.

Note that no assumptions are made about nodes stop-failing or new nodes joining the network. In fact, the solution described below can accommodate these situations and the computations adapt to the changes.

3.1. *Solution outline*

The solution for computing the robustness metric uses a primitive for calculating sums in a distributed network that is inspired by the gossip-like mechanism presented in [32]. The algorithm described in [32] computes a sum of values distributed on the nodes of a network using a property of order statistics applied to a series of exponential random variables. A formal description of this algorithm is given in Algorithm 2.

Algorithm 1: PropagateMinVal(v, τ).

```
; /* $v$, $\tau$ - received value and time-to-live; /* $v_{local}$, $\tau_{local}$ - local value and time-to-live */
*/
; /* $T$ - maximum time-to-live, constant value */
; /* $\mathcal{C}$ - constant value, default to 0.5 */
; /* create temporary variables */
$(v_m, v_M) \leftarrow (\min(v, v_{local}), \max(v, v_{local}))$ ;
$(\tau_m, \tau_M) \leftarrow$ corresponding $(\tau, \tau_{local})$ to $(v_m, v_M)$ ;
; /* update logic */
if $v_m == v_M$ then
    if $v_m < 0$ then                         /* equal negative values */
        $\tau_m \leftarrow \mathcal{C}\tau_m$
    else                                      /* equal positive values */
        $\min(\tau_m, \tau_M) \leftarrow \max(\tau_m, \tau_M) - 1$
else
    if $v_m < 0$ then                /* at least one negative value */
        if $v_m == -v_M$ then
            $(\tau_m, \tau_M) \leftarrow (T, T)$
        else
            $(\tau_m, \tau_M) \leftarrow (\mathcal{C}\tau_m, \mathcal{C}\tau_M)$
    else                          /* two different positive values */
        $\tau_M \leftarrow \tau_m - 1$
; /* update local variables */
$(v, v_{local}) \leftarrow (v_m, v_m)$ ;
$(\tau, \tau_{local}) \leftarrow$ corresponding $(\tau_m, \tau_M)$ ;
```

Algorithm 2: ComputeSum ($\mathbf{v}$, $\boldsymbol{\tau}$).

```
; /* $\mathbf{v}^0$ - original random samples vector on this node
*/
; /* $\mathbf{v}, \boldsymbol{\tau}$ - received value and time-to-live vectors */
; /* $T$ - maximum time-to-live, constant value */
; /* update all elements in the data vector */
for $j = 1$ to $length(\mathbf{v})$ do
    PropagateMinVal($\mathbf{v}[j]$, $\boldsymbol{\tau}[j]$)
; /* time-to-live update - do once every timeslot */
for $j = 1$ to $length(\mathbf{v})$ do
    if $\mathbf{v}[j] == \mathbf{v}^0[j]$ then          /* reinforce a minimum */
        $\boldsymbol{\tau}[j] \leftarrow T$
    else
        $\boldsymbol{\tau}[j] \leftarrow \boldsymbol{\tau}[j] - 1$;        /* decrease time-to-live */
        if $\boldsymbol{\tau}[j] <= 0$ then                  /* value expired */
            $\mathbf{v}[j] \leftarrow \mathbf{v}^0[j]$ ;
            $\boldsymbol{\tau}[j] \leftarrow T$
; /* estimate the sum of elements */
$s \leftarrow 0$ ;
for $j = 1$ to $length(\mathbf{v})$ do
    $s \leftarrow s + abs(\mathbf{v}[j])$
return $length(\mathbf{v})/s$
```

The algorithm resembles a gossip algorithm [19], but differs in a number of important points.

Essentially, the algorithm trades communications for convergence speed. By relying on the propagation of an extreme value (minimum value in this case) that is locally computable, it achieves the fastest possible convergence in a distributed network—$O(D\log N)$ time steps (where D is the network diameter and N is the number of nodes). This speed is significant compared with the original gossip algorithms that converge in $O(D^2\log N)$ time steps [9].

Fig. 1 illustrates the sum computation process under network dynamics. A network with 1000 nodes is modeled as a geometric random graph with diameter 14 and initialized with random values. Half of the network is disconnected at time step 50 and the nodes change their values at time step 200. The network values converge after 15 computation steps.

The price paid is the increased message size $O(\delta^{-2})$, where the parameter δ specifies the precision of the final result. Assuming that s is the ground-truth result, the algorithm offers an estimate in the interval $[(1-\delta)s, (1+\delta)s]$ with error $\epsilon = O(1/poly(N))$.

The extreme value propagation mechanisms are extended to account for network dynamics. Specifically, a time-to-live field is added to each value – the integer value decreases with time and marks the age of the current value. This mechanism takes care of nodes leaving the network, stop-crashing or resetting. In the example in Fig. 1, after convergence, half of the nodes in the network were removed at time step 50. The effect of the expiring time-to-live (set to a maximum of 50 in the example) can be seen around time step 100.

The time-to-live expiry mechanism is extended to achieve the removal of values in $O(D\log N + \log T)$ time steps. In other words, if a certain extreme value propagates through the network, it is marked as "expired" and its associated time-to-live value expires (i.e., reaches zero) within $O(D\log N + \log T)$ time steps. This is shown in Fig. 1 during the interval 200–300. At time step 200, half of the nodes in the network change their values randomly, triggering the expiration mechanism. A formal description of the extension is provided in Algorithm 1.

The distributed approach solves most of the scaling issues and is highly robust to network dynamics (e.g., network nodes becoming unavailable due to failures, reconfiguration, new nodes joining the system, etc.). As shown below, the approach is very fast for a typical network, significantly outperforming centralized approaches. Because the protocols rely on anonymous data exchange, privacy concerns [30] are alleviated; in any case, the identities of the system nodes are not needed in the computations.

The disadvantages of the approach map to the known properties of this class of epidemic algorithms. Although anonymity is preserved, an authentication system [20] is needed to prevent malicious data from corrupting the computations. Also, a light form of synchronization [41] is needed to coordinate nodes to report major changes to their local values. Clearly, the choice of an appropriate synchronization mechanism must take security into account [37].

Please cite this article as: Martijn Warnier et al., Distributed monitoring for the prevention of cascading failures in operational power grids, International Journal of Critical Infrastructure Protection (2017), http://dx.doi.org/10.1016/j.ijcip.2017.03.003

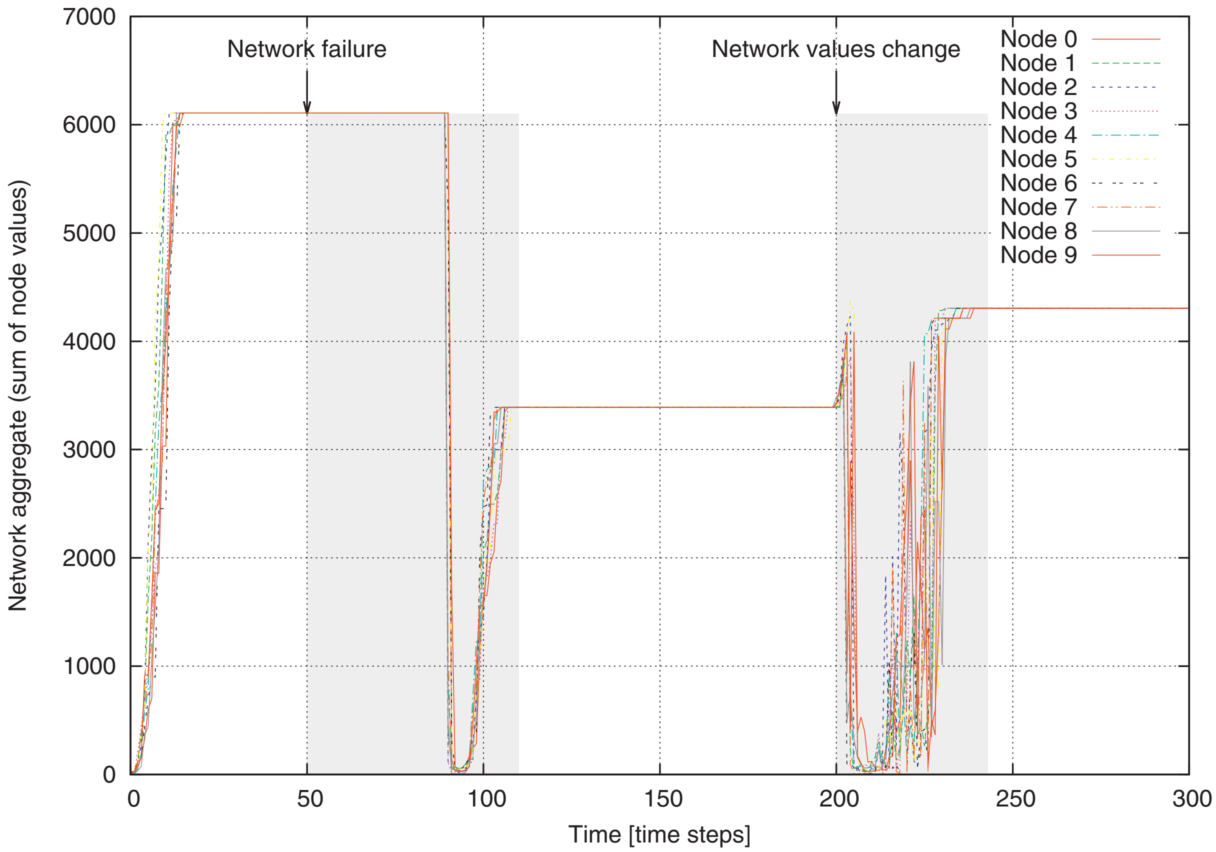


Fig. 1 – Sum computation process under network dynamics.

3.2. *Self-stabilizing sum computation*

The basic mechanism behind the sum computation algorithm is minimum value propagation via gossiping. Assume that each node holds a positive value x_i. At each time step, every node chooses a random neighbor and they exchange their values, both the nodes keeping the smallest value. The smallest value propagates rapidly in the network in $O(D\log N)$ time steps via this push–pull gossiping mechanism (see [36]). Algorithm 2 presents a formal description of the self-stabilizing sum computation.

Assume that each node i in a network holds a positive value x_i. In order to compute the sum of all n values in the network $\left(\sum_{i=1}^{N} x_i\right)$, Mosk-Aoyama and Shah [32] propose that each node holds a vector $\mathbf{v}$ of m values, initially drawn from a random exponential random distribution with parameter $\lambda_i = x_i$. After a gossiping step between two nodes i and j, the vectors $\mathbf{v}_i$ and $\mathbf{v}_j$ become equal, and hold the minimum value in each position of the initial vectors. Thus, given an index $k \in (1, m)$, the resulting vectors $\mathbf{v}'_i$, $\mathbf{v}'_j$ have the property:

$$\mathbf{v}'_i[k] = \mathbf{v}'_j[k] = \min\left(\mathbf{v}_i[k], \mathbf{v}_j[k]\right) \tag{6}$$

Mosk-Aoyama and Shah [32] show that, after all the vectors converge to some value $\mathbf{v}$, the sum of x_i values in the network may be approximated as:

$$\sum_{i=1}^{N} x_i = \frac{m}{\sum_{k=1}^{m} \mathbf{v}[k]} \tag{7}$$

This work extends the algorithm presented in [32] by adding, to each node, a new vector $\boldsymbol{\tau}_i$ that holds a time-to-live counter for each value. This new vector is initialized with a default value T, larger than the convergence time of the original algorithm (the choice of a proper value is explained below). The values in $\boldsymbol{\tau}_i$ decrease by one every time step, with one exception. The node generating the minimum $\mathbf{v}_i[k]$ in position $k \in (1, m)$ sets $\boldsymbol{\tau}_i[k]$ to T (Line 10 in Algorithm 2). In the absence of other dynamics, all the properties proved in [36] remain unchanged because the output of the proposed approach is identical to the original algorithm.

The main reason for adding the time-to-live field is to account for nodes leaving the network and nodes that fail-stop. This eliminates the requirement that nodes must keep track of their neighbors. Additionally, the proposed mechanism does not make use of node identifiers.

The intuition behind this mechanism is that a node that generates the network-wide minimum in position $k \in (1, m)$ will always advertise it with the accompanying time-to-live set to the maximum T. The remaining nodes adopt the value $\mathbf{v}[k]$ and have a value $\boldsymbol{\tau}[k]$ that decreases with distance from

Table 1 – Value propagation.

Propagation	Ordering	Previous	Intermediate	Final
None	$\mathbf{u}[k] < \mathbf{v}_i[k] < \mathbf{v}'_i[k]$	$\mathbf{u}[k]$	$\mathbf{u}[k]$	$\mathbf{u}[k]$
	$\mathbf{u}[k] < \mathbf{v}'_i[k] < \mathbf{v}_i[k]$	$\mathbf{u}[k]$	$\mathbf{u}[k]$	$\mathbf{u}[k]$
Slow	$\mathbf{v}_i[k] < \mathbf{u}[k] < \mathbf{v}'_i[k]$	$\mathbf{v}_i[k]$	$\mathbf{v}_i[k]$	$\mathbf{u}[k]$
	$\mathbf{v}_i[k] < \mathbf{v}'_i[k] < \mathbf{u}[k]$	$\mathbf{v}_i[k]$	$\mathbf{v}_i[k]$	$\mathbf{v}'_i[k]$
Fast	$\mathbf{v}'_i[k] < \mathbf{u}[k] < \mathbf{v}_i[k]$	$\mathbf{u}[k]$	$\mathbf{v}'_i[k]$	$\mathbf{v}'_i[k]$
	$\mathbf{v}'_i[k] < \mathbf{v}_i[k] < \mathbf{u}[k]$	$\mathbf{v}_i[k]$	$\mathbf{v}'_i[k]$	$\mathbf{v}'_i[k]$

the original node. T is chosen to be larger than the maximum number of gossiping steps it takes for the minimum to reach any node in the network. In a gossiping step between nodes i and j, if $\mathbf{v}_i[k] = \mathbf{v}_j[k]$, then the largest of $\boldsymbol{\tau}_i[k]$ and $\boldsymbol{\tau}_j[k]$ propagate (Line 13 in Algorithm 1). This means that $\boldsymbol{\tau}[k]$ on all nodes is strictly positive as long as the node is online. If the node that generates the minimum value in position k goes offline, then all the associated $\boldsymbol{\tau}[k]$ values in the network steadily decrease (Line 12 in Algorithm 2) until they reach zero and the minimum is replaced by next smallest value in the network (Lines 13–15 in Algorithm 2). T time steps are required for the network to “forget” the value in position k. Fig. 1 shows the graphical impact of the $O(T)$ mechanism during the interval 50 to 150.

The second self-stabilizing mechanism targets nodes that change their values at runtime. Assume that a node changes its value x_i to x'_i at some time t. This change triggers a regeneration of its original samples from the exponential random variable $\mathbf{v}_i$ to $\mathbf{v}'_i$. Let k be an index with $k \in (1, m)$. Let $\mathbf{u}$ be the vector containing the minimum values in the network if node i does not exist. In order to understand the change that occurs when transitioning from x_i to x'_i, it is necessary to consider the relationships between the individual values $\mathbf{v}_i[k]$, $\mathbf{v}'_i[k]$ and $\mathbf{u}[k]$.

As seen in Table 1, if $\mathbf{u}[k]$ is the smallest of all the values, then no change propagates in the network. If $\mathbf{v}'_i[k]$ is the smallest value, then it propagates rapidly – in $O(D\log N)$ time steps – with the basic extreme propagation mechanism. If $\mathbf{v}_i[k]$ is the smallest value, then the value remains in the network until its associated time-to-live field expires. Because $T \gg D$ usually holds, a mechanism is added to speed up the removal of this value from the network.

The removal mechanism is triggered by the node owning the value that needs to be removed (node i in this case). This is accomplished by node i marking the value $\mathbf{v}_i[k]$ as “expired” by propagating a negative value $-\mathbf{v}_i[k]$. This change does not affect the extreme value propagation mechanism (Algorithm 1) nor the estimation of the sum (note the use of the absolute value function in Line 19 in Algorithm 2). If node i contacts a node that also holds the value $\mathbf{v}_i[k]$, then it propagates the negative sign for the value and maximizes its time-to-live field to a large value T. Intuitively, as long as $\mathbf{v}_i[k]$ is present in the network, $-\mathbf{v}_i[k]$ propagates, causing it to be overwritten. Considering the large range of unique floats or double numbers versus the number of values in a network at a given time, it is safe to assume that the values in the network are unique.

The time-to-live field of any negative value halves with each gossiping step (for $\mathcal{C} = 0.5$) if it does not meet the $\mathbf{v}_i[k]$ value (Lines 11, 19 in Algorithm 1). Intuitively, if a negative value is surrounded by values other than $\mathbf{v}_i[k]$, it propagates while simultaneously canceling itself at an exponential rate. This mechanism resembles a predator–prey model [2] where the predators are represented by $-\mathbf{v}_i[k]$ and the prey by $\mathbf{v}_i[k]$. The mechanism is designed such that the populations cancel each other, targeting the fixed point at the origin as the solution for the accompanying Lotka–Volterra equations.

Lemma 1 (Value removal delay). *When using the value removal algorithm, the new minimum propagates in the network in $O(D\log N + \log T)$ time steps.*

Proof. In the worst case scenario, the entire network contains the minimum value $\mathbf{v}_i[k]$ in position k, with the time-to-live field set to the maximum T. The negative value, being the smallest value in the network, propagates in $O(D\log N)$ in the entire network. Again, in the worst case scenario, each node in the network has the value $-\mathbf{v}_i[k]$ in position k with the time-to-live set to the maximum T. From this moment on, the time-to-live halves at each gossip step at each node (for $\mathcal{C} = 0.5$), reaching zero in the worst case scenario in $O(\log T)$ time steps. This is the worst case because nodes may be contacted by several neighbors during a time step leading to much faster cancellation. Overall, the removal mechanism is active for at most $O(D\log N + \log T)$ time steps. This bound is an upper bound. In reality, the spread and cancellation mechanisms act in parallel, leading to tighter bounds. □

Lemma 1 provides the basis for choosing the constant T. Ideally, T should be as small as possible, in line with the diameter of the network. The fact that the removal mechanism is affected only by $\log T$ enables an overestimate of T to be used; the overestimate can be a few orders of magnitude larger than the diameter of the network with little impact on the convergence speed. For example, if the network diameter is 10 to 30 and the values refresh every 10,000 time steps, then T can be safely set to be anywhere in the range 1000–10,000 (see Section 4.3). This does not affect the convergence of the sum computation mechanism, but it allows for timely node removal.

All the mechanisms presented in this section lead to the sum computation mechanism *ComputeSum*() presented in Algorithm 2. It has the same properties as the original algorithm [32], but it incorporates self-stabilization properties that account for network dynamics in the form of node removal and nodes changing their values in batches.

3.3. *Robustness metric computation*

The robustness metric comprises two terms that can be computed locally (p_i in Eq. (2) and $R_{n,i}$ in Eq. (3)) and two that can be computed in a distributed manner (δ_i in Eq. (4) and R_{CF} in Eq. (5)). In particular, Eq. (5) can be rewritten as:

$$R_{CF} = \frac{\sum_{i=1}^{N} R_{n,i} P_i}{\sum_{j=1}^{N} P_j} \tag{8}$$

leading to a solution with two *ComputeSum*() algorithms in parallel. The first algorithm computes $\sum_{i=1}^{N} R_{n,i} P_i$ while the second algorithm computes $\sum_{j=1}^{N} P_j$.

Please cite this article as: Martijn Warnier et al., Distributed monitoring for the prevention of cascading failures in operational power grids, International Journal of Critical Infrastructure Protection (2017), http://dx.doi.org/10.1016/j.ijcip.2017.03.003

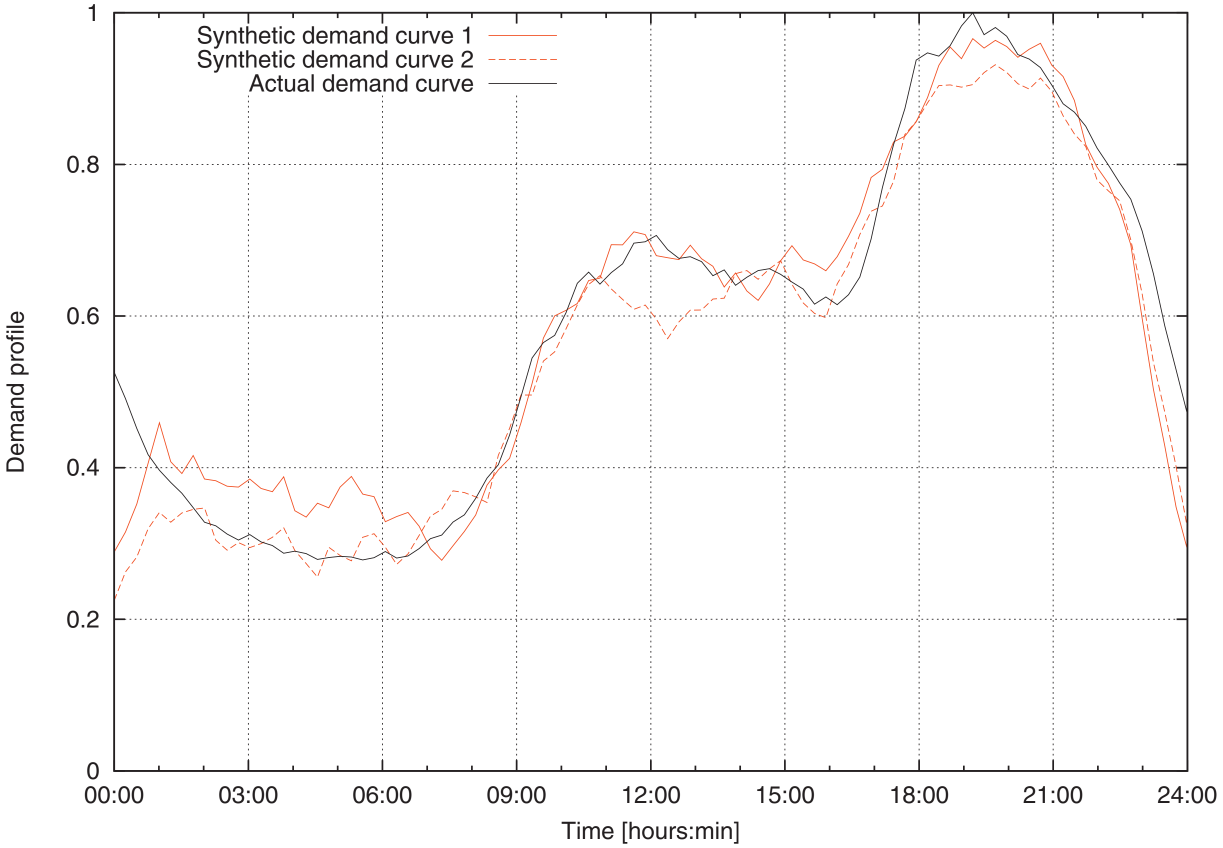


Fig. 2 – Actual demand profile in the Dutch transmission grid and two synthetically-generated demand profiles.

Characterizing the convergence time of the composition of two distributed algorithms is a difficult task in general. Fortunately, the composition of the two *ComputeSum*() algorithms has a convergence time equal to each of the two algorithms, leading to the same $O(D\log N + \log T)$ time steps complexity. If the network is stabilized, then after the power distributions P_i change, the values $\sum_{i=1}^{N} R_{n,i}P_i$ and $\sum_{j=1}^{N} P_j$ stabilize in $O(D\log N + \log T)$ in parallel, because they do not require intermediate results from each other. Because the gossip algorithms used in this work are based on minimum value propagation, all the nodes in the network have the same value after the algorithm converges. Stabilization is easily detected locally by monitoring the lack of changes in the propagated values for a fixed time threshold.

4. Analysis and discussion

The proposed approach for computing the robustness metric is scalable and robust. This section focuses on some of quantitative aspects and analyzes the results obtained from simulations based on synthetic and real data. The computer code implementing the proposed approach was implemented in Matlab and C++. In all the simulations, the network nodes were deployed in a square area. Their communications ranges were varied to obtain the desired value of the network diameter. Networks made up of several independent clusters were discarded.

4.1. *Data generation*

No public data is available that describes the structure and changes in load over time for a power grid. Therefore, data was generated in order to demonstrate the effectiveness of the proposed approach.

Computing the robustness of a power grid requires data describing its topology (i.e., interconnections of nodes with lines), electrical properties of its components (i.e., admittance values of transmission lines), information about its nodes (i.e., number and their types) and their generation and load values. The IEEE Power Systems Test Case Archive [12] provides all this data. In particular, the IEEE 118 Power System maintained in the archive provides a realistic representation of a real-world power transmission grid comprising 118 nodes and 141 transmission lines. The IEEE 118 system is used in this work as the reference power grid.

The IEEE 118 Power System includes information about the topology of the power grid. The loading profile provided with the grid topology gives a representative load for the network, but only for one moment in time. However, in practice, the topology of a power grid is generally unchanged over time (except for grid maintenance, failures and extensions) while the generation/loading profiles vary over time. The changing nature of the loading profile and, accordingly, the generation profile result in varying robustness over time. Therefore, simulating the robustness profile of a power grid over 1 day requires a demand profile for the entire day.

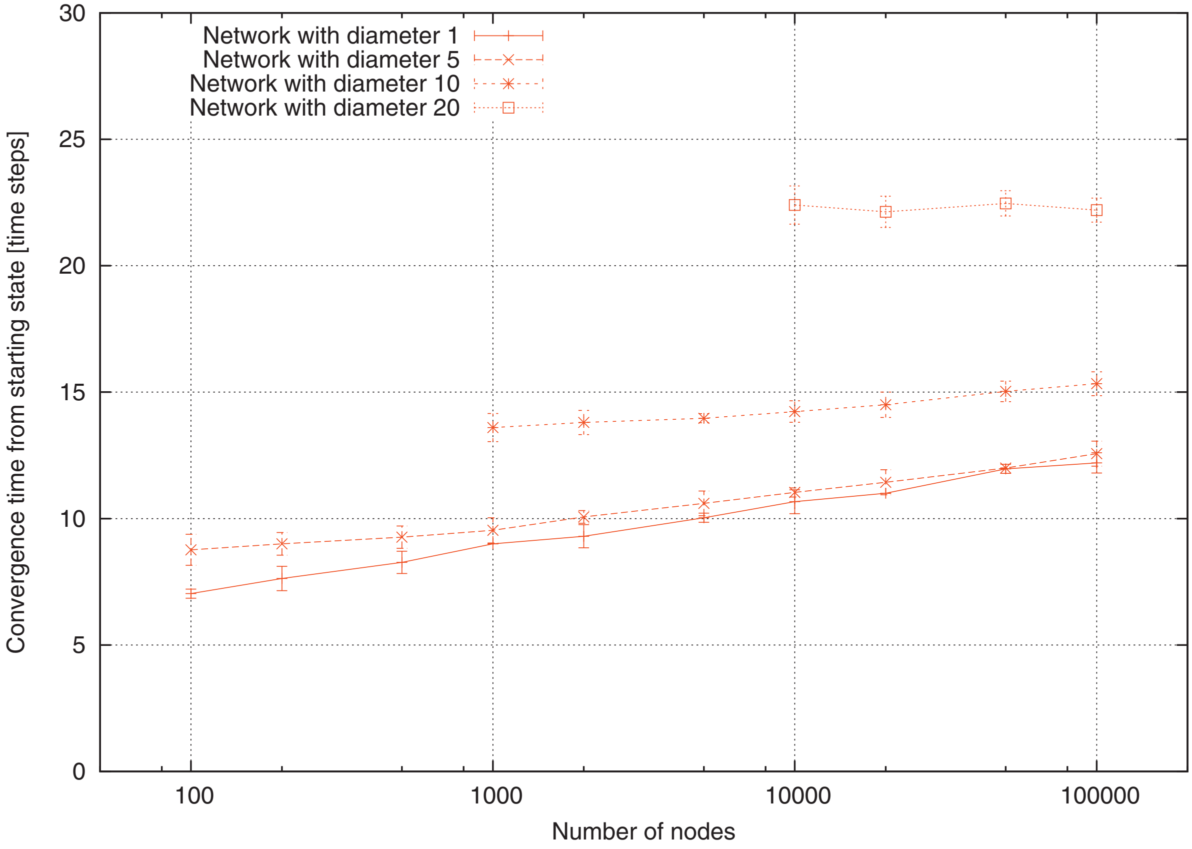


Fig. 3 – Convergence of the network starting from a clean state.

To vary the robustness of the IEEE 118 system, 10% of the power generation nodes were randomly chosen to be fed with synthetic (generated) demand profiles. The demand values of the other power generation nodes remained unchanged. The demand profiles were generated based on the actual load profile of the Dutch grid on January 29, 2006. The demands at the corresponding points in the Dutch grid were sampled every 15 min over the entire day.

Fig. 2 shows the actual demand profile at a point in the Dutch transmission grid and two synthetically-generated demand profiles. Each synthetic demand profiles was generated by introducing random noise to the actual demand profile and smoothing the curve using a moving average [21] with a window size of 10.

4.2. *Influence of communications topology*

The communications network underlying a smart grid can be implemented in a number of ways, each mapping to a possibly different communications topology. For example, an Internet backbone model that allows any-to-any communications in the network could be chosen, leading to a fully-connected graph.

In the first experiment, the network was modeled as a geometric random graph with its nodes initialized with random values, and the time taken for the aggregated sum to converge to the same value on all the nodes was recorded. Fig. 3 shows that the fastest aggregate computation occurs in a fully-connected network.

In the second experiment, the network was again modeled as a geometric random graph with its nodes initialized with random values. However, after the network stabilized, the values of half of the nodes were changed to different values. The time taken for the network to stabilize after this change was recorded. As expected, Fig. 4 shows that a fully-connected network stabilizes the fastest after a disruption.

These results assume that the Internet backbone works perfectly and is able to route the large amount of generated traffic. A more realistic scenario is that the data collection points obtain data from the individual consumers via some radio technology (e.g., GPRS modems) that are themselves connected to the Internet backbone. To keep the traffic in the network to a minimum, the data collection points only communicate with their first-order network neighbors, corresponding to a mesh network deployment. As seen in Figs. 3 and 4, the diameter of the network clearly has a major impact on the results; this confirms the theoretical convergence results. The information needs at least $O(D)$ time steps to propagate through the network. The constant in the $O()$ notation is influenced by the average network connectivity (a node can only contact a single neighbor per time step, slowing information dissemination) and by the push–pull communications model (a node may be contacted by several neighbors during a time step, speeding up information dissemination).

Please cite this article as: Martijn Warnier et al., Distributed monitoring for the prevention of cascading failures in operational power grids, International Journal of Critical Infrastructure Protection (2017), http://dx.doi.org/10.1016/j.ijcip.2017.03.003

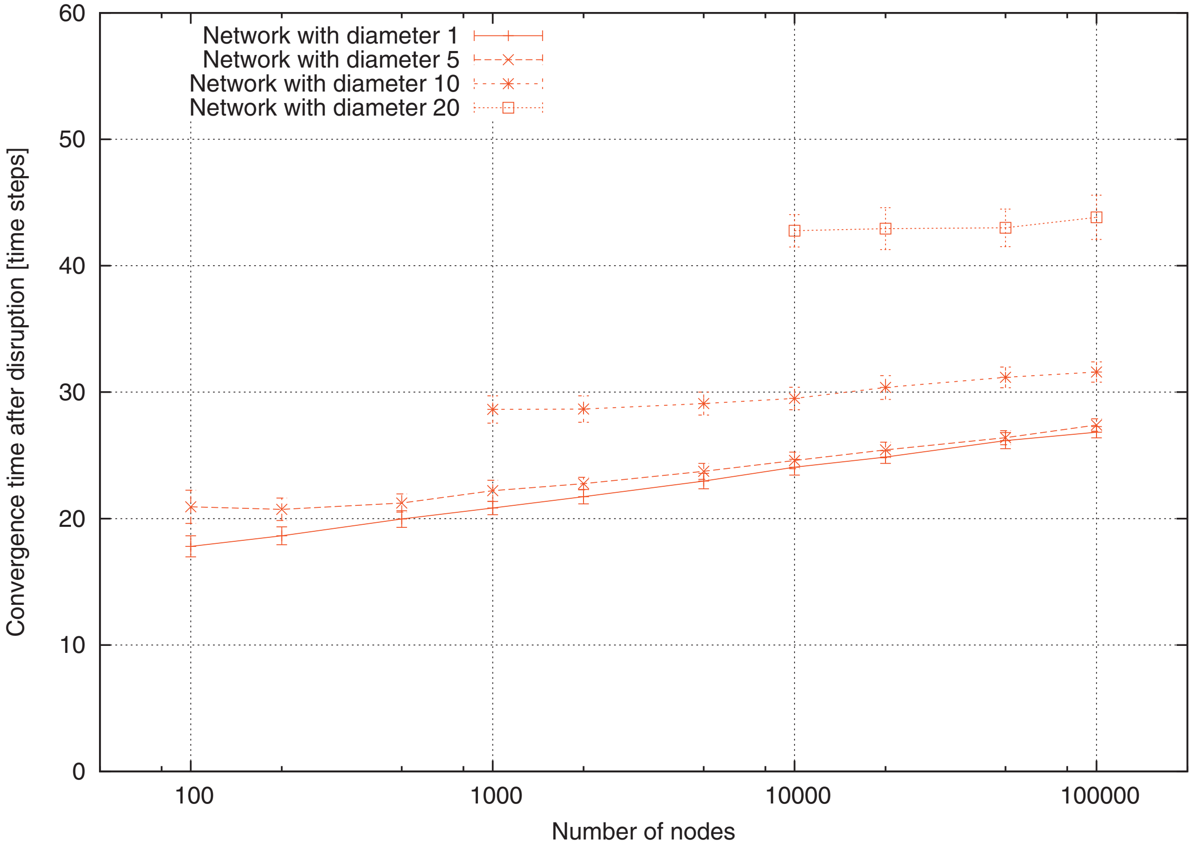


Fig. 4 – Convergence of the network after a disruption.

4.3. *Scalability aspects*

One of the main characteristics of the proposed approach is that the algorithm scales very well with the number of nodes in the network. As seen in Figs. 3 and 4, the number of nodes has little influence on the final results (influencing only as $O(\log N)$).

However, the simulations investigated cases where the number of nodes was varied over four orders of magnitude and the results hint that boundaries tighter than those proposed in this paper may exist. The results reveal that, in the case of a fully-connected network, the recovery time varies 34% between a network with 1000 nodes and one with 100,000 nodes, while the variation drops to just 2.4% for a 20-hop network varying from 1000 nodes to 100,000 nodes.

These results are very important for smart grid applications. Because the network is linked to a physical space (a country or, in general, a region) and fully covers the space, the diameter of the network is expected to, at most, decrease with the addition of new nodes. Intuitively, when thinking of nodes as devices with a fixed transmission range, adding more devices in the same region may lead to shorter paths. The aggregate computation approach proposed in this paper yields, on one hand, almost no variation for an increase in the number of network nodes and a linear variation with network diameter. These properties are essential to any solution that must take into account situations where the number of nodes in a grid increases over time.

Another experiment investigated the influence of the time-to-live of the negative fields on convergence and scalability. This experiment employed a random geometric graph of a 10-hop network with 1000 to 5000 nodes and varied the time-to-live for negative values between 500 and 10,000 (the values of half of the nodes were changed randomly after initial network convergence). The results in Fig. 5 confirm Lemma 1 with respect to the $\log T$ term. Specifically, the data shows that the convergence time is affected very little by the choice of parameters. As expected, the network diameter has a larger influence.

4.4. *Robustness metric computation*

Fig. 6 shows the performance of the distributed computation method on real data sets created in the manner described in Section 4.1. The figure shows the results of two simulation runs involving the distributed algorithms versus the results obtained by centralized computations (ground truth). The length of the value vector varies from 1000 values to 10,000 values and each point represents network data after convergence. The results confirm that the precision can be set to the desired value independent of the network topology and size.

When using a vector of 1000 elements, a mean relative error of 3% was obtained (maximum relative error 11% with a standard deviation of 2.6%). A larger vector (10,000 elements) produced a mean relative error of 1% (maximum relative error

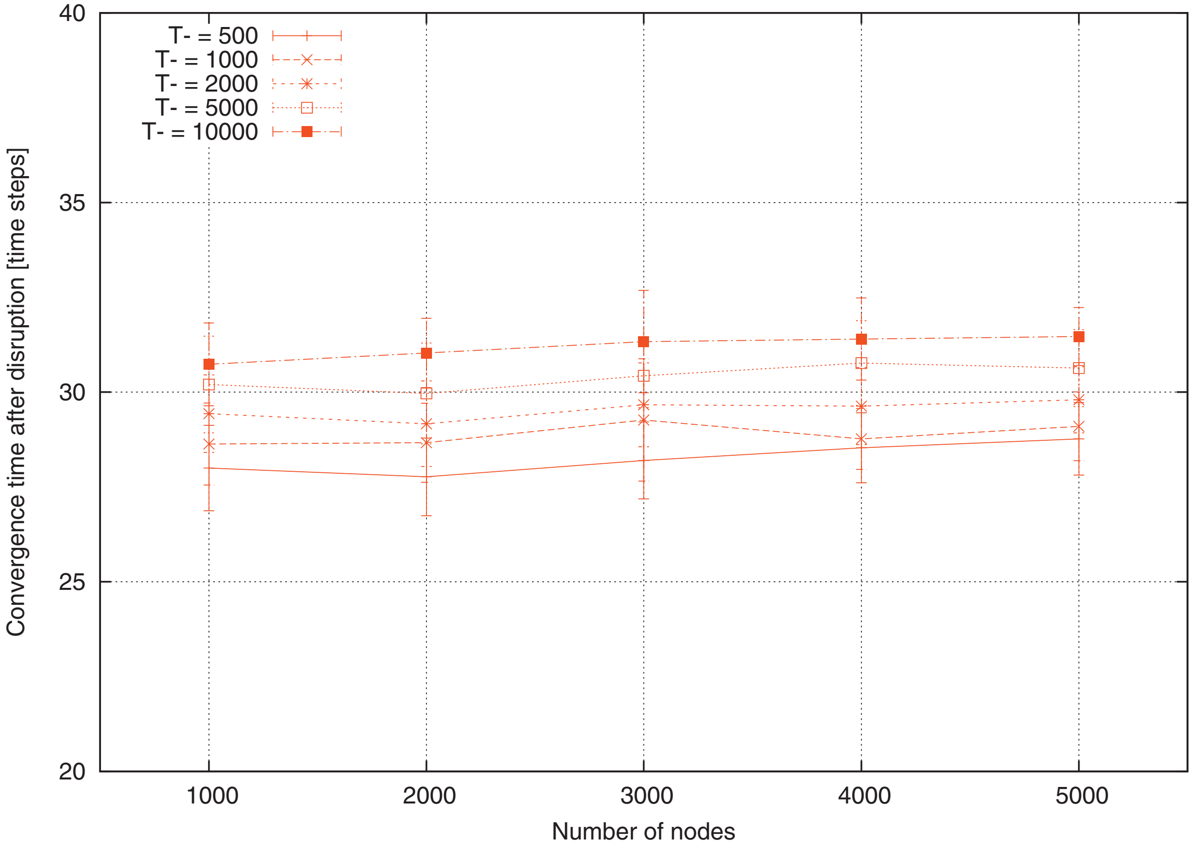


Fig. 5 – Influence of the *T* parameter.

of 4% with a standard deviation of 0.8%). These results are excellent, especially when considering that they were obtained from a combination of distributed computations with all the fault-tolerant mechanisms enabled.

Fig. 6 also incorporates a line at robustness value $R_{CF} = 0.67$ that corresponds to a critical threshold set by grid operators. If the robustness metric drops below this value, then a power line failure can lead to a blackout that affects more than 20% of the power grid. The threshold value was obtained by running cascading failure simulations of the IEEE 118 system using targeted attacks (i.e., worst-case scenario). Interested readers are referred to [27] for a structured methodology for determining such thresholds.

The critical threshold (which affects more than 20% of the power grid) is more or less arbitrary and was chosen purely for illustrative purposes. In practice, several factors have to be considered by grid operators (e.g., line capacities and maintenance cycles) to determine realistic threshold values. Nevertheless, the results illustrate the feasibility of the proposed approach because they clearly demonstrate that the error rate of the distributed algorithm is much lower than the minimal required drop in robustness needed to meet the threshold.

Finally, it is important to note that the proposed approach differs from traditional approaches that attempt to capture the global state of a network and then make decisions centrally (this is discussed in the next section). The proposed approach pushes the computations of the robustness metric in the network and the results are available at each node as soon as the computations converge. This mechanism can be readily used in the measurement phase, leading to the possibility of implementing distributed control loops on top of it.

5. Monitoring cascading failures in power grids

Research related to monitoring the state of a power grid can be divided into three main areas: (i) metrics that quantify the vulnerability of power grids to cascading failures; (ii) simulation models that predict the impacts of node/line outages; and (iii) sensor networks that capture the operative states of power grids.

A significant body of work exists on metrics for assessing the vulnerability of power grids to cascading failures. Most studies employ a purely topological or extended topological approach that primarily rely on graph-theoretic measures such as betweenness [40] or other centrality measures [37]. However, these studies (see, e.g., [7,8,11,22]) focus on the topological properties of power grids and fail to consider the operative states of the grids. In effect, this means that the metrics cannot be used to assess the changes in the vulnerability of an operational power grid.

Other researchers (see, e.g., [4,44]) have proposed measures that rely on simulation models. Although, these metrics incorporate the operative state of a power network in addition to its topology, it is very challenging to use the metrics to

Please cite this article as: Martijn Warnier et al., Distributed monitoring for the prevention of cascading failures in operational power grids, International Journal of Critical Infrastructure Protection (2017), http://dx.doi.org/10.1016/j.ijcip.2017.03.003

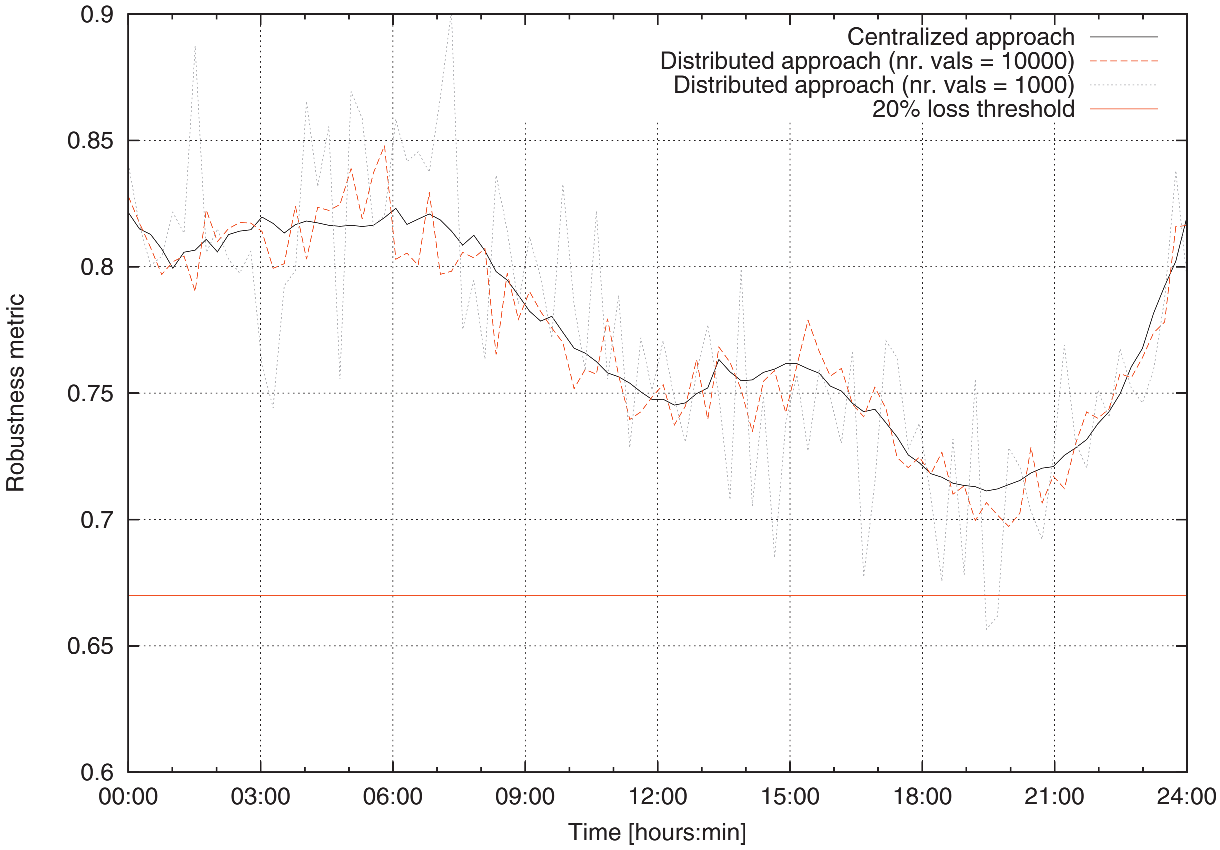


Fig. 6 – Robustness metric values.

quantify the resilience of the network to cascading failures in (near) real-time because the computations require full knowledge of the power grid state in order to simulate cascading failures. Previous work by the authors of this paper [24,25] is a notable exception. This is because it defines a metric that considers the topology and the operative state of a power grid while eliminating computationally-expensive tasks as in [31] or computing the complete network state in order to simulate cascading failures.

Grid operators traditionally assess network operations using flow-based simulation models (e.g., $N-x$ contingency analysis [38]). These models consider the operational behavior of a power grid. Grid operators can calibrate the models to match the power grid of interest and run various scenarios to assess the impacts of the failure of one or two lines. However, these models have two problems. First, they depend on the knowledge of grid operators who determine the failure scenarios that are to be explored. Second, the computational complexity of the simulation models render them infeasible in failure scenarios involving more than two components (i.e., $N-2$ contingency analysis). Interested readers are referred to [17] for an overview of contingency analysis methods for power grids.

Several schemes have been suggested for addressing the limitations of traditional contingency analysis methods. For example, Mittal et al. [31] have proposed a probabilistic contingency analysis scheme for power grids that allows contingency analysis up to eight levels deep (i.e., concurrent failures of up to eight nodes). Other researchers, such as Yan et al. [42], attempt to address the computational challenges by dividing a large power grid into smaller sub-grids (clusters) and running a separate contingency analysis on each of the smaller grids. By automatically adapting the sub-grid clusters over time based on different threat analysis scenarios and grid states, Yan et al. are able to perform detailed analyses of more realistic threats. Note that such approaches are generally better at dealing with more likely threat scenarios at the cost of ignoring less likely threat scenarios that might involve large impacts on a power grid. In contrast, the monitoring approach proposed in this paper considers all possible threats to the power grid at the same time based on the grid topology and state. This enables the determination of when a threat is more imminent without identifying the most vulnerable nodes. Thus, the approach complements contingency analysis approaches and could be used in conjunction with current grid operator practices.

Numerous researchers have specified distributed architectures for monitoring the state of a power grid. However, the architectures typically focus on data collection related to the loading levels of power lines, phase angles, etc. (see, e.g., [3,16,18,28,33,43,45,46]). Unfortunately, they do not use sophisticated data aggregation mechanisms to quantify the resilience of a power grid to cascading failures.

In conclusion, as far as the authors of this paper are aware, no power grid monitoring approaches are available that can – in near real-time – assess the vulnerability of an operational power grid to cascading failures.

6. Conclusions

This paper has introduced a novel distributed computation approach for assessing the resilience of an operational power grid to cascading failures in near real-time. The approach leverages a class of fast gossip algorithms [9] with self-stabilizing mechanisms for handling run-time network dynamics. The effectiveness of the approach is demonstrated by computing the robustness to cascading failures metric [24,25] in a rapid and reliable manner for a case study involving the benchmark IEEE 118 Power System.

The distributed computation approach has a number of desirable properties, most notably scalability and robustness. Simulation results performed with real and synthetic data demonstrate that the approach achieves very fast convergence times, influenced primarily by the network diameter and only logarithmically by the number of network nodes. This property is very important in smart grids because the numbers of nodes deployed in grids that cover regions or countries are expected to increase over the next few decades.

The precision of the computations can be set by adjusting the size of the messages exchanged in a network. This is a crucial property for scalability because message size is not a function of the number of network nodes. More importantly, the computation error scales as $O(1/poly(N))$, meaning that the greater the number of network nodes, the lower the final error. Finally, the approach preserves anonymity because it does not require unique identifiers for the network nodes.

The principal result is that it is possible to compute complex aggregates of the operational states of power grid nodes in a fully distributed manner rapidly (in near real-time) and reliably. Because automatic control systems always require measurements, the proposed approach is a perfect candidate for the measurement block of an automated distributed control scheme. While this paper has focused on the measurements of network properties, future research will investigate the actuation portion that is triggered by the availability of different power grid metrics.

Acknowledgments

This work was partially supported by the NWO Project, RobuSmart: Increasing the Robustness of Smart Grids through Distributed Energy Generation: A Complex Network Approach (grant no. 647.000.001) and by Rijksdienst voor Ondernemend Nederland (grant no. TKISG01002 SG-BEMS).

REFERENCES

[1] R. Albert, I. Albert and G. Nakarado, Structural vulnerability of the North American power grid, *Physical Review E*, vol. 69(2), 025103, 2004.

[2] R. Arditi and L. Ginzburg, Coupling in predator–prey dynamics: Ratio-dependence, *Journal of Theoretical Biology*, vol. 139(3), pp. 311–326, 1989.

[3] D. Bakken, C. Hauser, H. Gjermundrod and A. Bose, Towards More Flexible and Robust Data Delivery for Monitoring and Control of the Electric Power Grid, Technical Report EECS-GS-009, School of Electrical Engineering and Computer Science, Washington State University, Pullman, Washington, 2007.

[4] Z. Bao, Y. Cao, G. Wang and L. Ding, Analysis of cascading failure in electric grid based on power flow entropy, *Physics Letters A*, vol. 373(34), pp. 3032–3040, 2009.

[5] V. Belevitch, Summary of the history of circuit theory, *Proceedings of the IRE*, vol. 50(5), pp. 848–855, 1962.

[6] N. Bicocchi, M. Mamei and F. Zambonelli, Handling dynamics in diffusive aggregation schemes: An evaporative approach, *Future Generation Computer Systems*, vol. 26(6), pp. 877–889, 2010.

[7] E. Bompard, R. Napoli and F. Xue, Analysis of structural vulnerabilities in power transmission grids, *International Journal of Critical Infrastructure Protection*, vol. 2(1–2), pp. 5–12, 2009.

[8] E. Bompard, R. Napoli and F. Xue, Extended topological approach for the assessment of structural vulnerability in transmission networks, *IET Generation, Transmission and Distribution*, vol. 4(6), pp. 716–724, 2010.

[9] S. Boyd, A. Ghosh, B. Prabhakar and D. Shah, Gossip algorithms: Design, analysis and applications, *Proceedings of the Twenty-Fourth Annual Joint Conference of the IEEE Computer and Communications Societies*, vol. 3, pp. 1653–1664, 2005.

[10] B. Carreras, V. Lynch, I. Dobson and D. Newman, Critical points and transitions in an electric power transmission model for cascading failure blackouts, *Chaos*, vol. 12(4), pp. 985–994, 2002.

[11] X. Chen, Q. Jiang and Y. Cao, Impact of characteristic path length on cascading failure of power grid, *Proceedings of the International Conference on Power System Technology*, 2006.

[12] Department of Electrical Engineering, University of Washington, Power Systems Test Case Archive, Seattle, Washington (www2.ee.washington.edu/research/pstca), 2016.

[13] I. Dobson, J. Chen, J. Thorp, B. Carreras and D. Newman, Examining criticality of blackouts in power system models with cascading events, *Proceedings of the Thirty-Fifth Hawaii International Conference on System Sciences*, 2002.

[14] A. El-Sakkary, The gap metric: Robustness of stabilization of feedback systems, *IEEE Transactions on Automatic Control*, vol. 30(3), pp. 240–247, 1985.

[15] L. Freeman, A set of measures of centrality based on betweenness, *Sociometry*, vol. 40(1), pp. 35–41, 1977.

[16] A. Grilo, P. Gao, W. Xu and M. de Almeida, Load monitoring using distributed voltage sensors and current estimation algorithms, *IEEE Transactions on Smart Grid*, vol. 5(4), pp. 1920–1928, 2014.

[17] A. Gomez-Exposito, A. Conejo and C. Canizares (Eds.), *Electric Energy Systems: Analysis and Operation*, CRC Press, Boca Raton, Florida, 2016.

[18] V. Gungor, B. Lu and G. Hancke, Opportunities and challenges of wireless sensor networks in the smart grid, *IEEE Transactions on Industrial Electronics*, vol. 57(10), pp. 3557–3564, 2010.

[19] M. Jelasity, A. Montresor and O. Babaoglu, Gossip-based aggregation in large dynamic networks, *ACM Transactions on Computer Systems*, vol. 23(3), pp. 219–252, 2005.

[20] G. Jesi, D. Hales and M. van Steen, Identifying malicious peers before it's too late: A decentralized secure peer sampling service, *Proceedings of the First International Conference on Self-Adaptive and Self-Organizing Systems*, pp. 237–246, 2007.

[21] J. Kenney and E. Keeping, Moving averages, in *Mathematics of Statistics, Part 1*, J. Kenney (Ed.), Van Nostrand, Princeton, New Jersey, pp. 221–223, 1962.

[22] C. Kim and O. Obah, Vulnerability assessment of power grid using graph topological indices, *International Journal of Emerging Electric Power Systems*, vol. 8(6), article 4, 2007.

Please cite this article as: Martijn Warnier et al., Distributed monitoring for the prevention of cascading failures in operational power grids, International Journal of Critical Infrastructure Protection (2017), http://dx.doi.org/10.1016/j.ijcip.2017.03.003

[23] Y. Koc, T. Verma, N. Araujo and M. Warnier, MATCASC: A tool to analyze cascading line outages in power grids, *Proceedings of the IEEE International Workshop on Intelligent Energy Systems*, pp. 143–148, 2013.

[24] Y. Koc, M. Warnier, R. Kooij and F. Brazier, A robustness metric for cascading failures by targeted attacks in power networks, *Proceedings of the Tenth IEEE International Conference on Networking, Sensing and Control*, pp. 48–53, 2013.

[25] Y. Koc, M. Warnier, R. Kooij and F. Brazier, An entropy-based metric to quantify the robustness of power grids against cascading failures, *Safety Science*, vol. 59, pp. 126–134, 2013.

[26] Y. Koc, M. Warnier, R. Kooij and F. Brazier, Structural vulnerability assessment of electric power grids, *Proceedings of the Eleventh IEEE International Conference on Networking, Sensing and Control*, pp. 386–391, 2014.

[27] Y. Koc, M. Warnier, P. Van Mieghem, R. Kooij and F. Brazier, A topological investigation of phase transitions of cascading failures in power grids, *Physica A: Statistical Mechanics and Its Applications*, vol. 415, pp. 273–284, 2014.

[28] D. Li, F. Han and J. Xiao, Wide-area real-time dynamic security monitoring system of the North China power grid, *Power System Technology*, vol. 28(23), pp. 52–56, 2004.

[29] G. Masters, *Renewable and Efficient Electric Power Systems*, John Wiley and Sons, Hoboken, New Jersey, 2013.

[30] P. McDaniel and S. McLaughlin, Security and privacy challenges in the smart grid, *IEEE Security and Privacy*, vol. 7(3), pp. 75–77, 2009.

[31] A. Mittal, J. Hazra, N. Jain, V. Goyal, D. Seetharam and Y. Sabharwal, Real-time contingency analysis for power grids, *Proceedings of the Seventeenth International Conference on Parallel Processing*, vol. II, pp. 303–315, 2011.

[32] D. Mosk-Aoyama and D. Shah, Fast distributed algorithms for computing separable functions, *IEEE Transactions on Information Theory*, vol. 54(7), pp. 2997–3007, 2008.

[33] K. Moslehi and R. Kumar, A reliability perspective of the smart grid, *IEEE Transactions on Smart Grid*, vol. 1(1), pp. 57–64, 2010.

[34] A. Motter and Y. Lai, Cascade-based attacks on complex networks, *Physical Review E*, vol. 66(6), 065102, 2002.

[35] A. Pruteanu and S. Dulman, LossEstimate: Distributed failure estimation in wireless networks, *Journal of Systems and Software*, vol. 85(12), pp. 2785–2795, 2012.

[36] D. Shah, Gossip algorithms, *Foundations and Trends in Networking*, vol. 3(1), pp. 1–125, 2009.

[37] G. Stergiopoulos, P. Kotzanikolaou, M. Theocharidou and D. Gritzalis, Risk mitigation strategies for critical infrastructures based on graph centrality analysis, *International Journal of Critical Infrastructure Protection*, vol. 10, pp. 34–44, 2015.

[38] Task Force on Understanding, Prediction, Mitigation and Restoration of Cascading Failures, IEEE Power and Energy Society Computer and Analytical Methods Subcommittee, Vulnerability assessment for cascading failures in electric power systems, *Proceedings of the IEEE Power and Energy Society Power Systems Conference and Exposition*, 2009.

[39] M. Vaiman, K. Bell, Y. Chen, B. Chowdhury, I. Dobson, P. Hines, M. Papic, S. Miller and P. Zhang, Risk assessment of cascading outages: Part I – Overview of methodologies, *Proceedings of the IEEE Power and Energy Society General Meeting*, 2011.

[40] P. Van Mieghem, *Performance Analysis of Communications Networks and Systems*, Cambridge University Press, Cambridge, United Kingdom, 2006.

[41] G. Werner-Allen, G. Tewari, A. Patel, M. Welsh and R. Nagpal, Firefly-inspired sensor network synchronicity with realistic radio effects, *Proceedings of the Third International Conference on Embedded Networked Sensor Systems*, pp. 142–153, 2005.

[42] J. Yan, Y. Zhu, H. He and Y. Sun, Multi-contingency cascading analysis of smart grid based on self-organizing map, *IEEE Transactions on Information Forensics and Security*, vol. 8(4), pp. 646–656, 2013.

[43] Y. Yang, D. Divan, R. Harley and T. Habetler, Power line SensorNet—A new concept for power grid monitoring, *Proceedings of the Power Engineering Society General Meeting*, 2006.

[44] M. Youssef, C. Scoglio and S. Pahwa, Robustness measure for power grids with respect to cascading failures, *Proceedings of the International Workshop on Modeling, Analysis and Control of Complex Networks*, pp. 45–49, 2011.

[45] S. Zanikolas and R. Sakellariou, A taxonomy of grid monitoring systems, *Future Generation Computer Systems*, vol. 21(1), pp. 163–188, 2005.

[46] H. Zhang and L. Lai, Monitoring system for smart grid, *Proceedings of the International Conference on Machine Learning and Cybernetics*, pp. 1030–1037, 2012.

[47] N. Zhang, T. Zhou, C. Duan, X. Tang, J. Huang, Z. Lu and C. Kang, Impact of large-scale wind farm connecting with power grid on peak load regulation demand, *Power System Technology*, vol. 34(1), pp. 152–158, 2010.