A Simple Diffusion Model for Assessing the Effects of Attacks on Networks Partitioned in Communities

José Moronta[1], Claudio Rocco[2]
jmoronta@usb.ve, croccoucv@gmail.com

[1] Departamento de Tecnología Industrial, Universidad Simón Bolívar, Caracas, Venezuela
[2] Facultad de Ingenería, Universidad Central de Venezuela, Caracas, Venezuela

Abstract: This paper presents a first approach based on a simple diffusion model for evaluating the security characteristics of a system exposed to a random or intentional attack. The physical system is modeled as a network that is previously partitioned in communities. A simple diffusion model, based on cellular automata, is employed to describe the dynamics of spread of the hazard through the network while a Monte Carlo simulation allows simulating the stochastic evolution of the hazard between the nodes of the network. The assessment includes the behavior of the diffusion on each community as well as on the system. Two criteria are used to assess the security characteristics of the network: the First Time to Affect a Community and the Time to Reach a Percentage of Nodes of a Community. Two examples, using topologies of real electric power systems, are used to illustrate the proposed approach.

Keywords: Cellular Automata; Communities; Diffusion Model; Monte Carlo Simulation.

1. Introduction

This paper presents the development of an approach for evaluating the security characteristics of a network system (e.g. for water supply, electric power or gas distribution) exposed to random failures or intentional “attacks”. The system is represented as a set of nodes interconnected by links. A hazard, e.g. a poison, is injected into a node of the network. As soon as the “attack” is completed, its harmful effects begin to propagate through the network, from node to node. The diffusion is evaluated using a simple basic epidemic model where each node has only two possible states: Susceptible (S) and Infected (I) (SI model) [1]. When a generic node *i* is hit by the hazard, it propagates the “attack” to an adjacent node j through link *ij*, with a given probability.

Several authors have analyzed the diffusion problem in single networks (or “monoplex” networks) [e.g., 2-5 among others]. Recently the problem has also been considered in the general framework of multilayered networks, i.e., a data structure made of multiple layers, where each layer is a monoplex network, interconnected in disparate ways, such as “such as multilayer networks, multiplex networks, interdependent networks, networks of networks and many others” [5].

This paper presents an approach that extends previous works [2] in two basic aspects. First it detects communities, that is, groups of nodes that are “relatively densely connected to each other but sparsely connected to other dense groups in the network” [6]. Nodes within the same community have close connections and the connections between communities are relatively sparse. The resulting network, i.e., the set of communities and their inter-community links could be considered as a special case of a multilayer network: the interconnected network.

Second, the diffusion process is recorded at each community. This fact could suggest different “failure criteria”, i.e., sets of rules to declare the failure of one or more communities, such as, the disconnection of a given percentage of nodes, the islanding of the community, among possible criteria. From here, protection schema could be suggested to enhance the robustness of the network under study.

To our best knowledge, this two-steps analysis has not been previously presented.

The remainder of this paper is organized as follows: Section 2 describes the main concepts associated to a) network and its partitioning in community; and b) the SI model. Section 3 discusses the approach proposed for assessing the diffusion while Section 4 provides results of experimentations. Section 5 presents the main conclusions and future works.

2. Basic Background

2.1 Network and Partition in Communities

Let G (N, A, W) represent a network. N represents the set of nodes ($|N|=n$), $A=\{(i,j)\}$ is the set of links ($|A|=m$) and $w_{ij} \in W$ represents a characteristic of the link between two interconnected nodes i and j, such as reliability, capacity, among others.

As described by Porter et al. [6], a *community* is a network structure consisting of a group of nodes that are “relatively densely connected to each other but sparsely connected to other dense groups in the network”. Nodes within the same community have close connections and the connections between communities are relatively sparse. To illustrate, Figure 1 [7] shows a network and three communities: Community A={1,2}; Community B={3,4,5} and Community C={6,7,8,9}.

An interesting point is that many complex systems (e.g., electric power systems) can be represented through the partition of a network in "small, densely connected groups of nodes" [7].

Community detection techniques could be also used to assess the vulnerability of the different communities, such as the set of important nodes or links that when disconnected causes the island of one or more communities [7]. For example, in Figure 1, links between communities A and C or between B and C are considered as important.

The studies on community structures require the topology of the network and in special applications, the weights of the links. Several approaches have been proposed to identify communities in a network (for further information on community detection, the interested reader is directed to a recent survey by Fortunato [8]. In this paper the algorithm Fast Modularity [9], able to consider weighted graphs, and available in library igraph, in the R platform (2015) is selected.

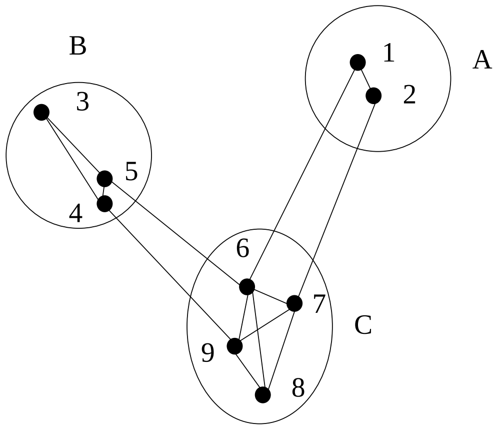


Figure 1: A Partition of Communities for a Network [7]

As mentioned earlier, in this paper the algorithm Fast Modularity [9] is used. The algorithm, a "bottom-up" agglomerative procedure, uses modularity (Q), an index for assessing the quality of the communities generated [10], to guide a greedy process.

The modularity, Q, associated to a set of k communities is defined by [10] as:

$$Q = \begin{smallmatrix} k \\ i=1 \end{smallmatrix} \; \frac{e_i}{m} - \; \frac{d_i}{2*m} \; ^{2} \tag{1}$$

where:
k defines the number of communities, e_i defines the numbers of links in community i, d_i is the sum of the degree of each nodes in community i (the node degree is defined as the number of links originating from a given node), and m is the total number of links in the network. The value of Q provides a measure of "the difference between the observed connectivity within communities and its expected value for a random graph with the same degree sequence" [11]

In the first step of the algorithm, each node is assigned to its own community. Then, an appropriate pair of nodes, connected by a link, is joined to create a new structure. The process is repeated until all nodes belong to a single community. To define the pair of nodes to merge, the procedure calculates Q for the current structure, and each possible merge is considered, with its corresponding Q. The one with the highest increment in Q is then selected as the new structure (the structure and Q are stored). At the end, the structure with the highest Q is selected as the best community structure. The main steps of these merging strategies are as provided by [7]:

1. Consider each node in the network as a single community;

2. Calculate the change in modularity, ΔQ_{ij} when creating a new community from communities i and j;

3. Combine the two communities with highest ΔQ_{ij};

4. Repeat steps 2 and 3 until $\Delta Q_{ij} \leq 0$.

The fast version of the algorithm uses efficient data structures, which allows a complexity of O($n\log^2 n$), on sparse graphs. To illustrate the process, the following sequences described in Table I are obtained by applying the algorithm to the network shown in Figure 1.

Table I: Illustrative Example of Community Detection for Network in Figure 1 [7]

k	**Communities Merged**		**Communities**	**Q_k**	**ΔQ_{ij}**
0	-	-	{1},{2},{3},{4}, {5},{6},{7},{8},{9}	-0.12245	-
1	{2}	{1}	{1,2},{3},{4}, {5},{6},{7},{8},{9}	-0.06122	0.061225
2	{3}	{4}	{1,2},{3,4},{5}, {6},{7},{8},{9}	-0.00510	0.056122
3	{3,4}	{5}	{1,2},{3,4,5}, {6},{7},{8},{9}	0.09949	0.104592
4	{8}	{7}	{1,2},{3,4,5}, {6},{7,8},{9}	0.14031	0.040816
5	{8,7}	{9}	{1,2},{3,4,5}, {6},{7,8,9}	0.21174	0.071429
6	{8,7,9}	{6}	{1,2},{3,4,5}, {6,7,8,9}	0.28571	0.073979
7	{1,2}	{8,7, 9,6}	{1,2,6,7,8,9}, {3,4,5}	0.26531	-0.020408

At the end of any community detection algorithm, the groups of nodes belonging at each community, as well as the set of Inter-Community Links (ICLs) are obtained.

2.2 The Susceptible and Infected Model

A simple model is used to mimic the diffusion of a given disruptive event in a node or set of nodes of the network under study. In this model, the status of each node is defined as susceptible (S) or infected (I). A node switches from S to I when the node is directed attacked or when any node in its neighbors has been previously infected. The diffusion is controlled by a parameter that represents the probability to transfer the hazard to its neighbors. This simple model has been used by Pepyne et al. [12] in the analysis of cascade effects on electric power systems. In this case, the disconnection of a node (i.e., an electric substation) forces the power flow to be redistributed among the nearby lines causing potential overload of specific components and triggering protecting devices to disconnect the lines with a probability of failure propagation β (as described in [12], in power systems this probability is a function of protection enhancements or maintenance actions).

Of course, other models based only on network topology could be used [13-16]. Also, it is possible to consider other simpler models, as the SIR model that includes an additional state to model a removal or a recovery [1]. In Rocco et al. [2] the authors included additional features such as a time delay or the possibility that a set of specific nodes could not be infected. However, they did not analyze network partitioned in communities.

In this paper, the diffusion process is simulated using a simple “cellular automata” (CA), as described in [17-18]. Within a CA computational scheme, each node *i* is mapped into a spatial cell whose neighborhood N_i is the set of cells (i.e. network nodes) which provide their input to it. The state variable s_i of cell *i* is binary, assuming the value of 1 when node *i* is infected (active) and of 0 when not operating (passive). The transition rule, governing the evolution of cell *i* consists of the application of the logic operator OR ($\vee$) to the states of the nodes in its neighborhood:

$$s_i(t+1) = s_p(t) \vee s_q(t) \vee ... \vee s_r(t) \qquad (2)$$

where *t* is the iteration step and $p, q, ..., r \in N_i$.

It is important to mention that in (2), the activation of a node is random and depends on the value of β. A high value of β means that the diffusion could be faster. In general, the process is repeated and the average values are reported.

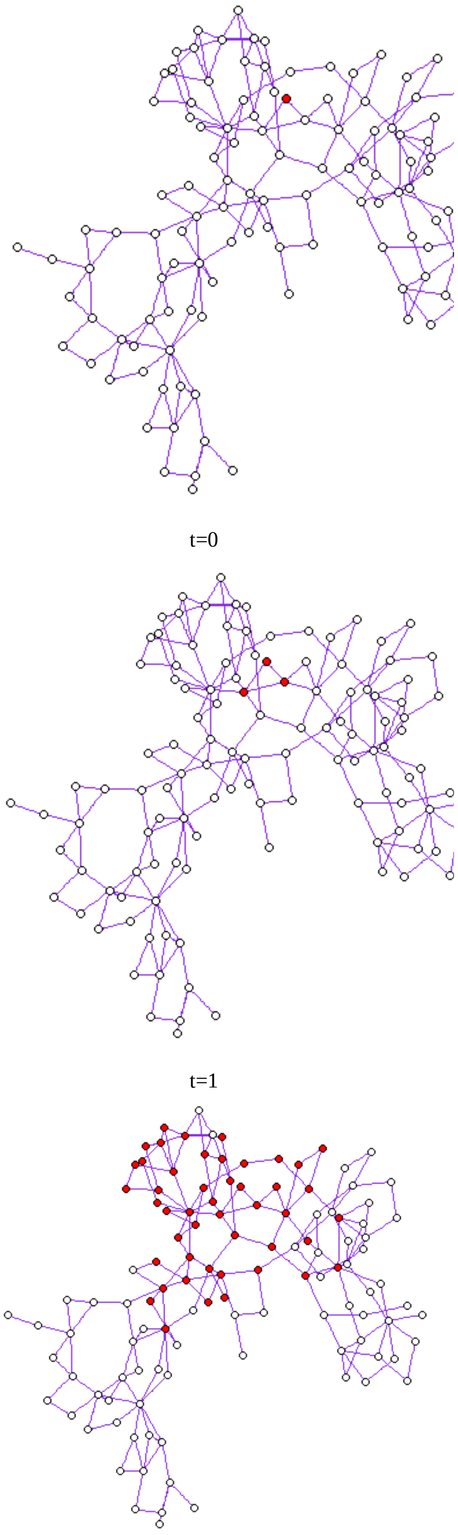
To illustrate the procedure, Figure 2 shows the topology of a network and the position of the node exposed to a disruptive event at *t*=0. Figure 2 also shows the evolution at selected times. At the end of the process, all of the nodes will be eventually infected.

3. Proposed Approach

The methodological approach proposed for evaluating the security characteristics of a partitioned network system exposed to random failures or intentional “attacks” consists on:

1. Detect C, the set of communities using a predefined set of weights W (|C|=K)
2. Given DE, the set of Disruptive Events, perform the diffusion process and measure different possible indexes, for each community, such as, the First Time to Affect a Community (FTAC$_i$, i=1,…, K) or the Time to Reach a Percentage of Nodes of a Community (TRP%NC$_i$, i=1,…, K), among others.

Step 2 could be repeated NSIMUL times to mimic the random nature of the diffusion process [19]. In this case the indexes derived are considered as random variables and probability estimations could be derived.



t=0

t=1

t=15

Figure 2: Time Evolution of the Diffusion Process at Selected Times

4. Case Studies

In this section, the topologies of two real electric power systems are analyzed. The following assumptions are used to illustrate the approach:

- The network topology is free of error
- NSIMUL is fixed at 1000.
- Community detection is performed using the algorithm Fast Modularity [9], available in library igraph 0.7.2 (in the R platform [20])
- The probability of failure propagation β is a unique value
- The diffusion process is repeated 100 times and average values are reported
- The weight vector is randomly generated in U[0,1] and represents a specific characteristic of each link

4.1 *Italian High-Voltage Electric Power System*

Figure 3a shows the network associated to the Italian 380 kV power transmission grid as described in [21]. The network consists of 127 nodes and 171 links. Figure 3b shows the set of 11 communities (K=11) detected from step 1 of the proposed approach (nodes belonging to a community have the same color). The ICLs are 25 and are represented in red.

Figure 4 shows the percentage of nodes failed in each community as a function of time derived from step 2 of the proposed approach based on the following assumptions: a) A single node is randomly selected (i.e., |DE|=1); b) the disruptive event is randomly selected among N; and c) β = 0.35.

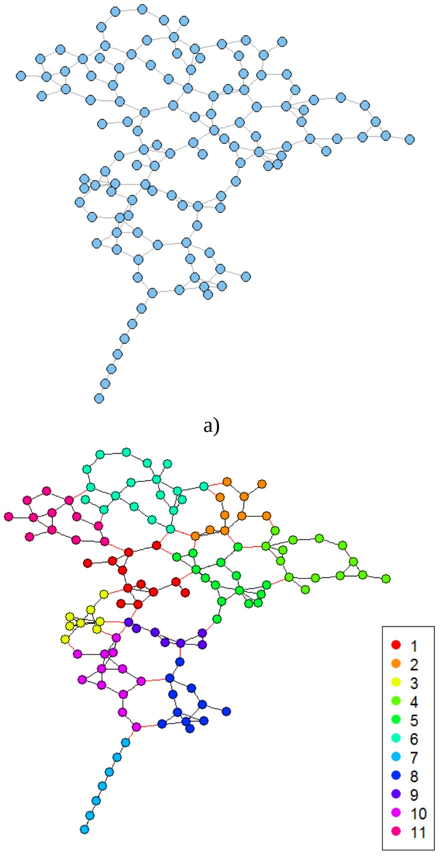
For example, Figure 4 shows that at time t=4, two communities are partially affected (communities 2 and 4) while others have no nodes affected. It also evidences that the communities reach the 100 % of infected nodes at different times. At t=44, the diffusion process stops.

From Figure 4, it is possible to obtain, for example, TRP%NC_i (the first time that all of the communities have a predefined percentage of nodes infected). For example, at TR60%NC_i = 37, all of the communities have at least 60 % of the nodes affected. It is important to realize that this time corresponds to a single evolution of the disruptive event considered.

In order to illustrate the random nature of the process, step 2 of the proposed approach is repeated NSIMUL=1000 times. Figure 5 shows the boxplot associated to TR60%NC_i. Considering the median values, at t=40 all of the communities have at least 60 % of the nodes affected.

Figure 6 shows the cumulative distribution associated to the $FTAC_i$. Note that for community 4, the plot only shows one point with probability 1, indicating that the disruptive event starts at a node in this community. From Figure 6 it is also clear that the last community affected is community 7.

Figure 7 shows the percentage of nodes failed in each community as a function of time when two disruptive events are selected among the set of all of the nodes. As expected, now TR60%NC_i = 28 is less than the value derived for a single disruptive event and the diffusion process stops at t=35.



a)

b)

Figure 3: The 380kV Italian Network:

a) Topology; b) Communities

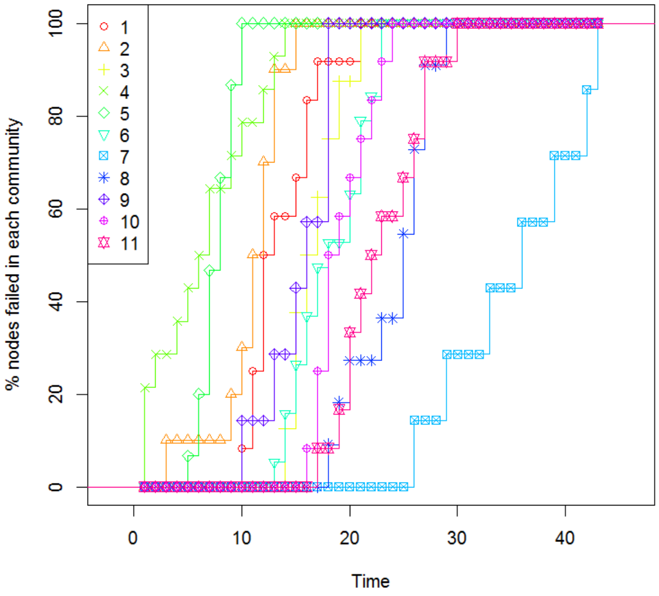


Figure 4: Time Evolution of the % of Nodes Affected in each Community. The Disruptive Event is Selected among all of the Nodes

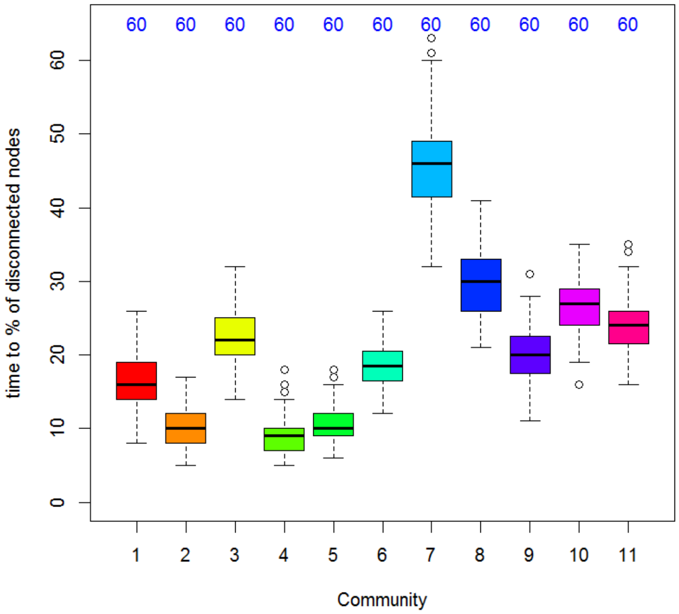


Figure 5: Time to Reach a 60% of Nodes Disconnected in each Community. The Disruptive Event is Selected among all of the Nodes

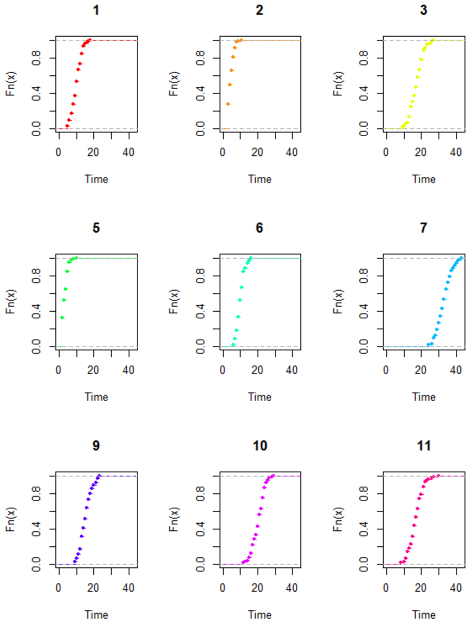


Figure 6: Cumulative Distribution of the First Time to Affect a Community. The Disruptive Event is Selected among all of the Nodes

Figure 8 shows the cumulative distribution associated to the FTAC$_i$. Note that the disruptive events belong to nodes in communities 3 and 4.

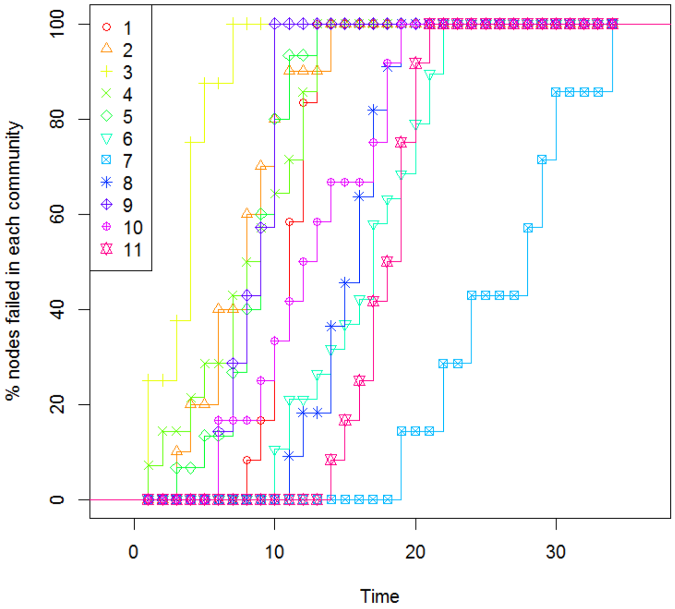


Figure 7: Time Evolution of the % of Nodes Affected in each Community. Two Disruptive Events are Selected among all of the Nodes

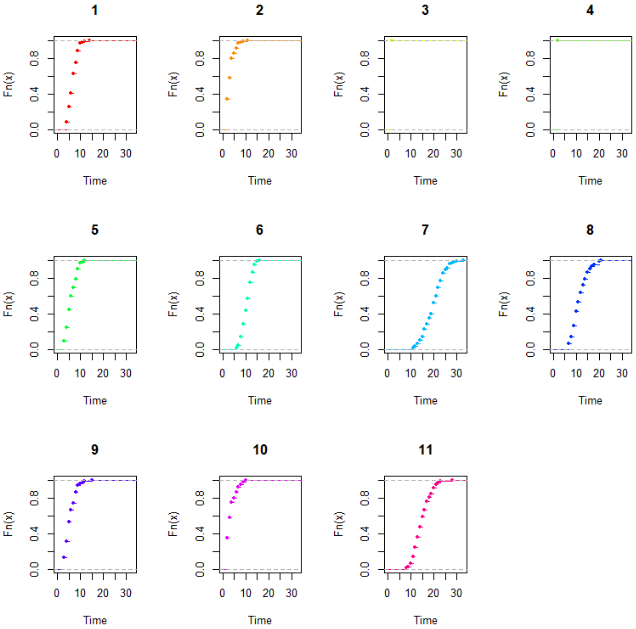


Figure 8: Cumulative Distribution of the First Time to Affect a Community. Two Disruptive Events are Selected among all of the Nodes

4.2 Venezuela Power System

The Venezuela Power System (known as SEN in Spanish) represents the high voltage system, from 138 to 765 kV [22]. The graph contains 139 nodes and 270 links. Figure 9 shows the topology and the ten communities detected (due to confidential reasons, the topology is shown without any geographic reference).

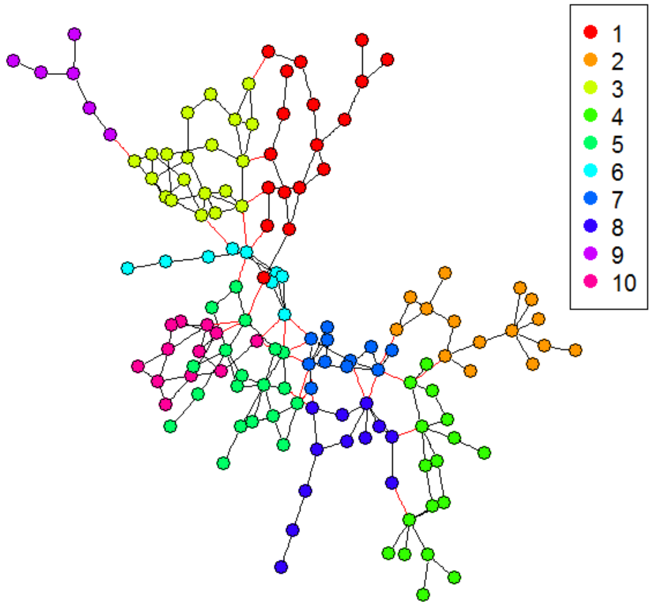


Figure 9: The Venezuela Power System

Figure 10 shows the percentage of nodes failed in each community as a function of time. In this network, the diffusion process stops at t=39. At TR60%NC$_i$=25, all of the communities have at least 60 % of the nodes affected.

Finally, Figures 11-12 show the boxplot associated to TR60%NC$_i$ and the cumulative distribution associated to the FTAC$_i$ respectively. Note that the disruptive event belongs to community 1 and, based on the median values, communities 2, 4 and 9 have the highest first time to be affected.

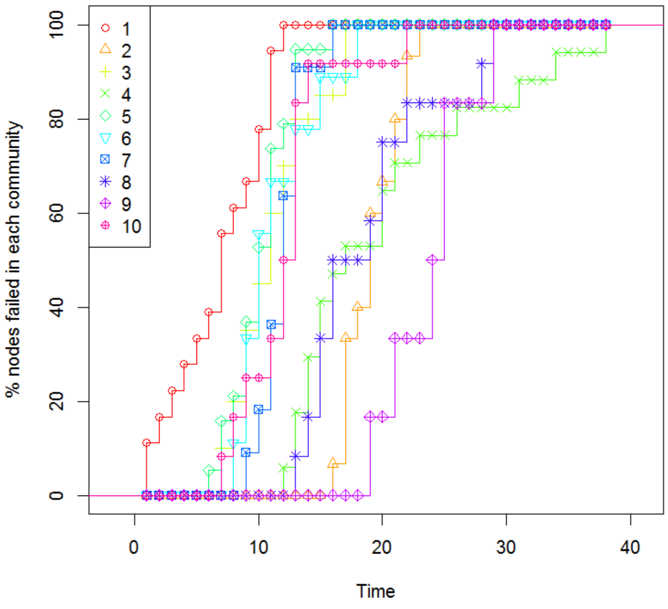


Figure 10: Time Evolution of the % of Nodes Affected in each Community. The Disruptive Event is Selected among all of the Nodes

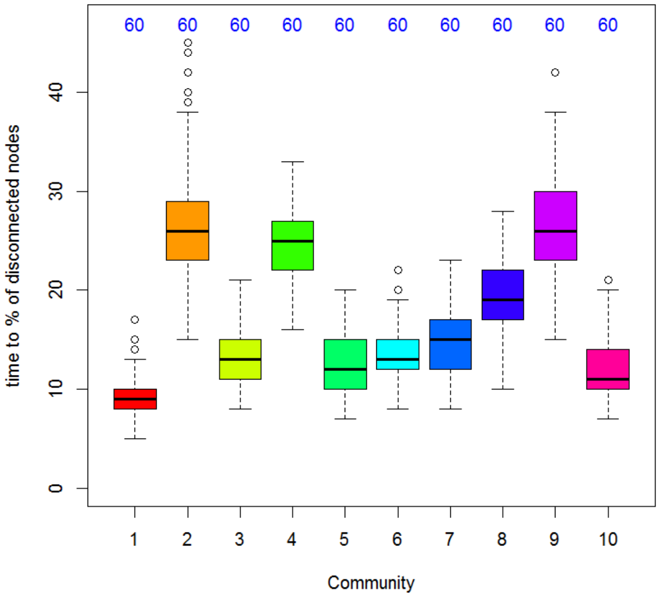


Figure 11: Time to Reach a 60% of Nodes Disconnected in each Community. The Disruptive Event is Selected among all of the Nodes

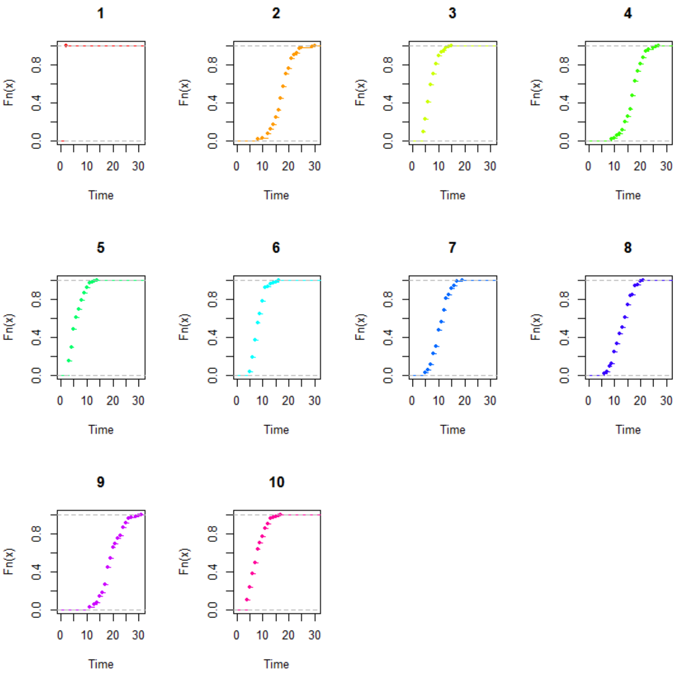


Figure 12: Cumulative Distribution of the First Time to Affect a Community. The Disruptive Event is Selected among all of the Nodes

5. Conclusions And Future Works

This paper presented an extension of a previous approach based on CA and MC simulations for assessing the security of a network by simulating how an “attack” is propagated through the system. The physical system is now modeled as a network partitioned in communities and the diffusion process is recorded at each community. This fact allows defining different "failure modes", i.e., different conditions for considering the malfunction of one or more communities or the whole system. To this aim, two indexes have been proposed to quantify: a) the time for the first impact in a community or b) the time for affecting a given percentage of nodes.

Yet, additional studies are required to derive the appropriate corresponding models to include, for example, the damage caused during the diffusion or including restoration actions or protective devices. The use of simulation models such as the ones here proposed could allow managing the attack emergency by identifying the most critical vulnerabilities, at a system or community level, that, for example, minimize the time for a selected percentage of nodes to be affected, and how to effectively handle attacks.

REFERENCES

[1] H. Hethcote, *The Mathematics of Infectious Diseases*, SIAM REVIEW, vol. 42, no. 4, pp. 599–653. 2000.

[2] C. Rocco, E. Zio, D. Salazar and G. Muller. *Vulnerability Assessment in Complex Networks Exposed to Terrorist Hazard: A Multiple-Objective Optimization Approach*. In Reliability and Maintainability Symposium, 2007. RAMS'07. IEEE, Florida, USA, January 2007. pp. 196-201.

[3] M. Kivelä, A. Arenas, M. Barthelemy, J. Gleeson, Y. Moreno and M. Porter. *Multilayer networks*. Journal of Complex Networks, vol. 2, no. 3, pp. 203–271. 2014.

[4] A. Garas. *Interconnected Networks*, Springer International Publishing Switzerland. ISSN 1860-0832. 2016.

[5] M. Salehi, R. Sharma, M. Marzolla, M. Magnani, P. Siyari, and D. Montesi. *Spreading Processes in Multilayer Networks*. IEEE Transactions on Network Science and Engineering, vol. 2, no. 2, pp. 65-83. 2015.

[6] M. Porter, J. Onnela, and P. Mucha. *Communities in Networks.* Notices of the American Mathematical Society, vol. 56, no. 9, pp. 1082-1097. 2009.

[7] C. Rocco and J. Ramirez-Marquez. *Vulnerability Metrics and Analysis for Communities in Complex Networks*, Reliability Engineering and System Safety, vol. 96, no. 3, pp. 1360-1366. 2011.

[8] S. Fortunato. *Community Detection in Graphs.* Physics Reports, vol. 486, no.75, pp. 75-174. 2010.

[9] A. Clauset, M. Newman and C. Moore. *Finding Community Structure in Very Large Networks*. Physical review E, vol. 70, no. 6, 066111. 2004.

[10] M. Newman and M. Girvan, *Finding and Evaluating Community Structure in Networks*. Physical review E, vol. 69, no. 2, 026113. 2004.

[11] M. Porter, J. Onnela and P. Mucha. *Communities in networks*. Notices of the AMS, vol. 56, no. 9, pp. 1082-1097. 2009.

[12] D. Pepyne, C. Panayiotou, C. Cassandras and Y. Ho. *Vulnerability Assessment and Allocation of Protection Resources in Power Systems*. In American Control Conference. IEEE, Virginia, USA, June 2001. pp. 4705-4710.

[13] A. Motter and Y. Lai. *Cascade-Based Attacks on Complex Networks*. Physical Review E, vol. 66, no. 6, 065102. 2002

[14] A. Motter. *Cascade Control and Defense in Complex Networks*. Physical Review Letters, vol. 93, no. 9, 098701. 2004

[15] P. Crucitti, V. Latora and M. Marchiori. *A Model for Cascading Failures in Complex Networks*. Physical Review E, vol. 69, no. 4, 045104. 2004

[16] Z. Wu, G. Peng, W. Wang, S. Chan and E. Wing-Ming. *Cascading Failure Spreading on Weighted Heterogeneous Networks*. Journal of Statistical Mechanics: Theory and Experiment, vol. 2008, no. 05, 05013. 2008

[17] S. Wolfram. *Origins of Randomness in Physical Systems*. Physical Review Letters, vol. 55, no. 5, 449. 1985.

[18] C. Rocco and E. Zio. *Solving Advanced Network Reliability Problems by Means of Cellular Automata and Monte Carlo Sampling*, Reliability Engineering and System Safety, vol. 89, no. 2, pp. 219-226. 2005.

[19] M. Marseguerra and E. Zio. *Basics of the Monte Carlo Method with Application to System Reliability*. LiLoLe, Verlag GmbH (Publ. Co. Ltd.) 2002.

[20] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL http://www.R-project.org/. 2015.

[21] P. Crucitti, V. Latora and M. Marchiori. *Locating Critical Lines in High Voltage Electrical Power Grids*. Fluctuation and Noise Letters, vol. 5, no. 02, pp. 201-208. 2005.

[22] C. Yajure, D. Montilla, J. Ramirez-Marquez and C. Rocco. *Network Vulnerability Assessment Via Bi-Objective Optimization with a Fragmentation Approach as Proxy*. Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability, vol. 227, no. 6, pp. 576-585. 2013.