CSS 2022 - Agent-based malware propagation model analysis

Viviane Desgrange, Charlotte Felius

June 2022

1 Agent-based malware propagation in complex network model

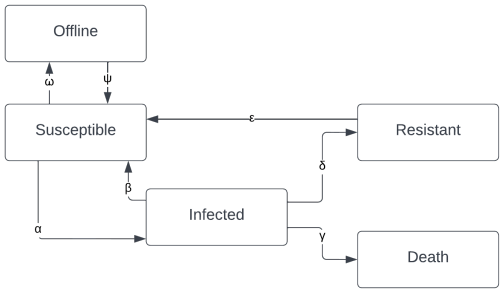


Figure 1: Agent-based model rules

The implemented model is based on the papers. It was implemented using the mesa framework for building, analysing and visualizing agent-based models. The complex network are generated using networkx graph () and provided to Mesa framework has a NetworkGrid component.

One agent is added to each node of the graph and associated with a list of parameters:

1. malware_spread_chance : Probability α to infect neighbor nodes.
2. recovery_chance : Probability β to recover.
3. death_chance : Probability γ to die.
4. gain_resistance_chance : Probability δ to became resistant to the malware right after recovering (β). IE. log4j library was updated to fix its vulnerability.

5. susceptible_chance : Probability ϵ for resistant node to be susceptible again. IE. library get a new vulnerability.

6. importance : Importance of the node. If too important, it cannot be offline. Probability ψ to get online.

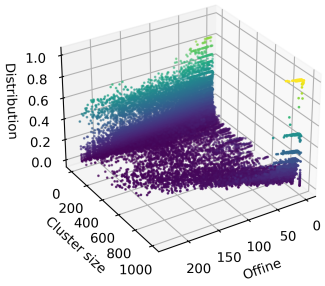
7. offline_probability : Computed during agent step. Probability ω to be offline (disconnected to not get infected) if the node is not too important.

8. malware_check_frequency : Probability to check its state during this step. Not everyone has its device fully secure, libraries might not be up-to-date despite issues, antivirus might not have check every files, etc.

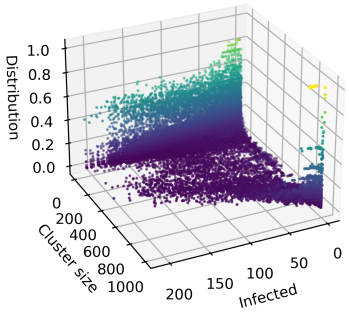
2 Cluster analysis in ABM-model

2.1 Experiment 1 - Analysis of cluster size distribution with regards to agent states

With this analysis, we are trying to check if any information can be inferred from the distribution of the cluster size with regards to the agent states. It was observed during early simulations than the number of clusters evolve through time during the simulation and especially depending of the number of agents in a specific states. The ideas was to check if some specific pattern emerge, if there were some giants components, etc.
Tested with 100 and 1000 nodes on 100 steps for 100 simulations on different networks. Initial outbreak perform on node of greatest degree.



(a) Cluster size distribution with regards to offline agents

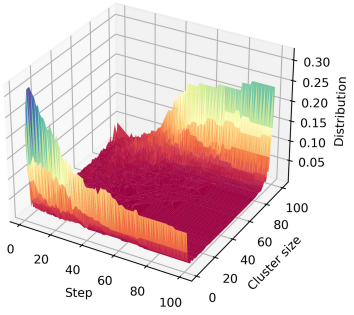


(b) Cluster size distribution with regards to infected agents

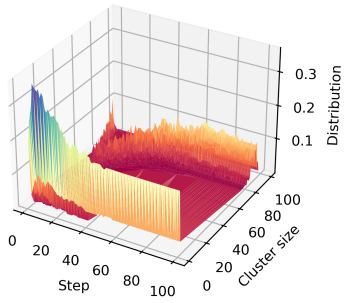
With no surprises, results showed that distribution of cluster size might change with the number of nodes offline or infected (which trigger offline nodes). With regards to the other agent states, nothing special could be observed. Experiment was stopped here.

2.2 Experiment 2 - Analysis of cluster size distribution with regards to time steps

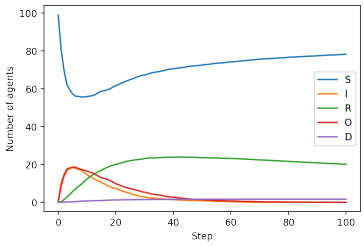
While experiment 1 shows nothing specific on distribution of cluster size with regards to the state of the agents, it might seems interested to observe the phenomena with regards to the time steps. Exclude the time step 0 where the network is fully connected, and single nodes.
Tested with 100 and 1000 nodes on 100 steps for 100 simulations on different networks. Initial outbreak perform on node of greatest degree.



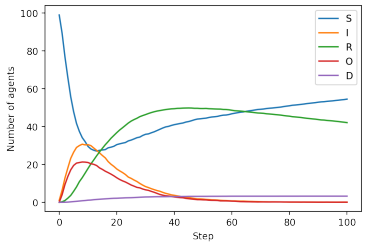
(a) Cluster size distribution with regards to simulation time steps. 100 nodes. Albert-Barabasi graph.



(b) Cluster size distribution with regards to simulation time steps. 100 nodes. Erdős–Rényi graph.



(a) Evolution of agent states. 100 nodes. Albert-Barabasi graph.



(b) Evolution of agent states. 100 nodes. Erdős–Rényi graph.

Results shows presence of a large number of cluster of small size on average at the beginning of the simulations, before eventually the emergence of giant components: the number of clusters of great size increase over time. This phenomena is observable for both type of graph but more especially on the Albert-Barabasi network.

We observed the emergence of giant component is correlated to the evolution of the agents' states, as the number of infected agent increase, dead and offline agents increase generating small cluster size. When infected agents decrease greatly, offline node eventually go back online and giant component emerge again.

2.3 Experiment 3 - Cascading failure (avalanche of number of clusters) in the network

The goal of this experiment is to look at the existence of avalanche in the number of clusters in the network. While experiment 1 shows nothing special on distribution of cluster size, it has been observed some avalanche in the number of clusters when their were some phase transition in the agent state.

H-I-R-E-D model got results on a similar hypothesis. Lack of time to run the experiment on this model, so experiment was dropped.

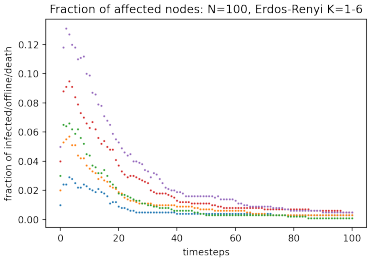
2.4 Experiment 4 - Analysis of cluster entropy

This experiment aim at studying the creation of communities (not disconnected sub-components) which might be connected by using an algorithm such as *Girvan–Newman method* and *Louvain method*. Then to analyse these communities, analysing entropy for instance.

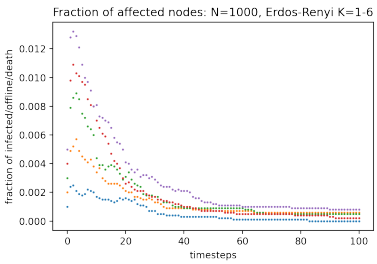
Lack of time to run the experiment on this model, so experiment was dropped.

3 Affected fraction of nodes per network size

Here, we calculated the average amount of affected nodes (infected, offline and death) for every timestep and multiple simulations from 100 timesteps and different amount of starting nodes with an infection (K). We observe that for a lower K, the average fraction of affected computers are initially higher if the K is higher, but mostly converge between 40-60 timesteps.



(a) ER: avg fraction of affected nodes calculated per timestep for N = 100



ER: avg fraction of affected nodes calculated per timestep for N = 1000

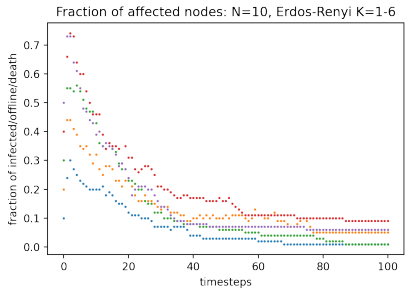


Figure 6: Erdos-Renyi: Average amount of fraction of affected nodes calculated per timestep for N = 10

4 SPYWARE: Antivirus Agent Analysis

If Spyware is noticed, an antivirus or software update is likely to be developed to clean up the spyware. Spyware is a bit different in the sense that it could get unnoticed for a long period of time, but antivirus software can be developed at any moment. However, we assume in this experiment that there is below 5 percent of chance that an antivirus is developed randomly, and when 60% of the nodes are either infected or offline, that an antivirus will be found with certainty due to large amount of infected computers.

Therefore, in this experiment we analyed after which timestep an antivirus would be deployed on average for different cluster sizes. The extensive results for this experiment are located on github in `SIROD/notebooks/antivirus.ipynb`.

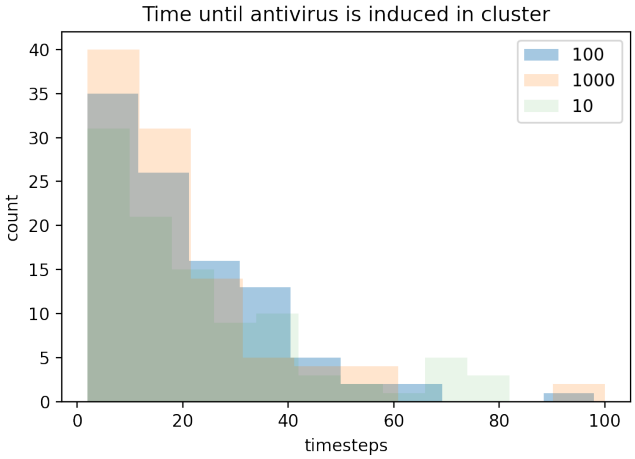


Figure 7: Distribution of timesteps until an Antivirus is found for different N

We notice that, remarkably, for larger cluster sizes it takes shorter on average

for spyware to get noticed by infected computers. This is due to the fact that in larger clusters, the spread of spyware will go faster, and more computers will notice in general that they have been infected. For every infected computer the chance is larger that an antivirus software is found, because more computers are warning that they have this virus. However, these results mean for an attacker that when a small cluster is attacked the spyware can get unnoticed for a large amount of time and they will succeed in gathering data for a longer period.