

## **Multi-agent simulations for virus propagation in D2D 5G+ networks**

Ziyad Benomar<sup>1</sup>, Chaima Ghribi<sup>1</sup>, Elie Cali<sup>1</sup>, Alexander Hinsen<sup>2</sup>,

Benedikt Jahnel<sup>2 3</sup>, Jean-Philippe Wary<sup>1</sup>

submitted: August 29, 2022

<sup>1</sup> Orange Labs  
44 Avenue de la République  
92320 Châtillon  
France  
E-Mail: ziyad.benomar@orange.com  
chaima.ghribi@orange.com  
elie.cali@orange.com  
jeanphilippe.wary@orange.com

<sup>2</sup> Weierstrass Institute  
Mohrenstr. 39  
10117 Berlin  
Germany  
E-Mail: alexander.hinsen@wias-berlin.de  
benedikt.jahnel@wias-berlin.de

<sup>3</sup> Technische Universität Braunschweig  
Universitätsplatz 2,  
38106 Braunschweig  
Germany

No. 2953  
Berlin 2022



---

2020 *Mathematics Subject Classification.* 60K35, 60F10, 82C22.

*Key words and phrases.* Agent-based modeling and simulation, malware spread, device-to-device.

We thank the team from Orange S.A for inspiring discussions. This work was funded by the German Research Foundation under Germany's Excellence Strategy MATH+: The Berlin Mathematics Research Center, EXC-2046/1 project ID: 390685689 as well as Orange Labs S.A.

Edited by  
Weierstraß-Institut für Angewandte Analysis und Stochastik (WIAS)  
Leibniz-Institut im Forschungsverbund Berlin e. V.  
Mohrenstraße 39  
10117 Berlin  
Germany

Fax: +49 30 20372-303  
E-Mail: [preprint@wias-berlin.de](mailto:preprint@wias-berlin.de)  
World Wide Web: <http://www.wias-berlin.de/>

# Multi-agent simulations for virus propagation in D2D 5G+ networks

Ziyad Benomar, Chaima Ghribi, Elie Cali, Alexander Hinsen,  
Benedikt Jahnel, Jean-Philippe Wary

## Abstract

In this paper we present results for an extended class of multi-agent simulation models for malware propagation in device-to-device 5G networks, first exhibited in [1]. The models allow to understand and analyze mobile malware spreading dynamics in highly dynamical networks and also to assess the effectiveness of a proposed counter measure policy for reversing attacks and securing the system. Our main simulation studies identify critical thresholds for maximal malware propagation and isolate two distinguished regimes for malware survival and extermination depending on a variety of parameters. We further predict via simulations the malware spreading velocities, depending on device density and speed, as well as the percentage of counter agents that have to be introduced into the network for malware elimination. We complement these findings and state also an associated theoretical study that highlights the key parameters of our agent-based model and exhibit certain linear relationships between them.<sup>1</sup>

## 1 Introduction

Device-to-device (D2D) communications is one of the key emerging technologies for 5G networks and beyond. It enables a direct exchange of data between mobile devices which extends coverage for devices lacking direct access to the cellular infrastructure and therefore enhances the network performance. However, security issues are very challenging for D2D systems as malware can easily compromise mobile devices and propagate across the decentralized network. Compromised devices represent infection threats for all of their connected neighbors as they can, in their turn, propagate malware through susceptible devices and form an epidemic outbreak. This enables attackers to infect a larger population of devices and to launch cyber- and physical malicious attacks. Therefore, it is of great importance to have a good understanding of vulnerability and security issues, particularly the malware propagation processes, in such networks and to be able to design optimal defence strategies against attacks.

Modeling malware propagation in D2D is challenging due to the complexity of such networks induced for example by topology for device mobility. In order to cope with this, D2D can be investigated and analyzed using analytical models (e.g., stochastic geometry, point processes, etc.). Some of these approaches have been proposed to model malware spreading in D2D networks [11, 22, 23]. Nevertheless, classical simulation and analytical tools are often not suitable for capturing the global dynamics of complex systems.

In this paper we propose to tackle the problem from the perspective of complex-systems science and present a new agent-based model (ABM) in order to analyze and understand malware propagation

---

<sup>1</sup>This is an extended version of the conference contribution [1] with additional material on counter measures. Other parts are reported unchanged.

in D2D networks. For this, the agent-based simulation approach provides the possibility to simulate complex-systems dynamics and to test theories about local behaviors and their emergence. Unlike traditional techniques of simulation, based on mathematical or stochastic models, multi-agent simulation is more suitable for complex problem modeling and simulation. In fact, applying classical simulation and analytical tools, such as differential equations, to complex systems often produces undesired complications. Indeed, many challenges that arise in the traditional numerical modeling come from the fact that individual actions (activities that result in a modification of the system) and their impact on the dynamics of the system are often underrepresented. Usually, individual behaviors, i.e., decisions made at the individual or group level, cannot be incorporated into these simulations. On the other hand, in a multi-agent simulation, the model is not a set of equations as in mathematical models, but a set of entities. Here agents represent the set of all the simulated individuals, objects encode the set of all represented passive entities, and the environment is the topological space where agents and objects are located and which they can move in and act upon.

Although agent-based simulations have been successfully used to model complex systems in different areas like biology, sociology, political science and economics, it is still insufficiently explored in the field of telecommunication networks, specifically for malware spreading in D2D. In this work, we aim to shed more light on whether such highly dynamical D2D networks can be treated as a complex system and whether complex-systems science can give insights on the emergent properties of malware propagation. The main contributions of this paper are as follows:

- We propose a new ABM for studying malware propagation in D2D 5G+ networks and we formally prove its correctness for predicting different properties of agents over the time horizon.
- We present a theoretical study to estimate the critical values of the model's parameters and to identify the most important ones to consider for simulations.
- We exhibit a variety of simulations that analyze malware spreading dynamics. In particular, we report on results about critical thresholds for maximal malware infection and for the separation of survival and extermination regimes. Further we examine important characteristics such as the malware infection rate and velocity under a change of parameters.
- Additionally, we exhibit simulation results on counter measure efficiency for the control of malware propagation. In particular, we estimate the percentage of anti-malware devices needed in the network in order to eliminate the virus.

The rest of the manuscript is organized as follows. Section 2 reviews related work. Section 3 describes the ABM for malware propagation in D2D networks. Section 4 shows details of our multi-agent simulation implementation. Section 5 presents a theoretical study for the problem in some specific scenarios. Section 6 shows simulation results followed by conclusions in Section 7.

## 2 Related work

ABMs are effective and robust tools in simulating complex and dynamic phenomena like epidemic spreading. These models have been used primarily in epidemiological studies of infectious diseases and have recently gained a great importance also in the epidemiological modeling as can be seen from the vast literature in the context of the COVID-19 pandemic, see for example [2, 7, 17, 18].

However, ABMs are still in their infancy with regard to telecommunication networks. Let us mention that some ABMs have been proposed in the literature for IoT networks [4, 13, 21]. Other applications

of ABMs to telecommunication networks are proposed in [19] and [20], where the authors analyze the effectiveness of ABMs to understand self-organization in peer-to-peer and ad-hoc networks. These studies provide further motivation to our investigation on applying ABMs for the study of malware spreading dynamics and possible counter measures in D2D 5G networks.

Let us again note that conventionally D2D systems are modeled using analytical methods (e.g., stochastic geometry) which have proven to be powerful tools for modeling spatial device and road systems. For example, in several recent works dealing with D2D modeling (see e.g., [5, 15]), streets are represented by random tessellations. Then, devices are placed on the streets according to Poisson point processes, thus forming Cox point processes (see for instance [6]). In this context, for example shadowing, being a critical feature for 5G in urban environments, can be efficiently encoded by a line-of-sight connectivity as described and studied in [16].

Let us further mention that the authors in [22, 23] present a framework for the modeling and analysis of malware spread in D2D with mobile devices and study some strategies of both defenders and attackers. The proposed model is based on an analytical approach and does not consider urban environments. In view of this, a standard SIR model, as presented in [12], studies malware propagation in D2D considering urban environments but mobility is not taken into account. Even though the obtained results were promising, some questions remained open regarding the convergence of the malware propagation speed, the shape theorem of the infection and the critical thresholds. This mainly comes from the fact that the dynamics of the system were insufficiently captured.

In relation to this, our prior work [11] studies malware propagation and counter measures in D2D networks also in urban environments. Here we present a stochastic-geometry-based model that takes into account street systems as well as shadowing effects (important for 5G) in urban environments (as proposed in [16]) but does not consider device mobility.

### 3 Multi-agent models for virus propagation in D2D

In this section we give a detailed description of the D2D malware-propagation model in urban environments. For this, we will first introduce the different model layers as a dynamical stochastic-geometry system, before rephrasing the model in the language of multi-agent systems. We stress that in this ABM description, devices are represented as reactive agents that move in the environment and have a variety of capabilities like neighborhood discovery, malware propagation and malware removal.

In short terms, the system has the following composition. We first consider an urban environment. Then, at initial time, devices are placed randomly on the streets, where we make the simplifying assumption that devices that are situated in buildings are not to be taken into account. This can be justified by the high frequencies used in 5G. The devices move independently and randomly at a constant speed. Moreover, two devices can communicate directly with each other if they are close enough and on the same street. Let us note that this approach takes shadowing into account, but not interference. At time zero, a virus is introduced carried by a device near to the center of the city. The virus can now propagate from one device to another if they can communicate for a long enough time that represents both the discovery time plus the transmission time. In the final step, following [12], we will introduce devices called *white knights*, carrying a patch that is able to erase the virus from attacking devices if under attack, and then transmit this white-knight ability to the attacker.

### 3.1 Stochastic geometry for communication systems

As already mentioned, our model is composed of several different layers: First, we model the street system, then we model the initial placement of devices and their mobility, then we define the connection graph, then the infection rules, and then, optionally, we choose some part of the devices to be white knights.

#### 3.1.1 Street systems and devices

We consider our urban street environment  $E$  as a two-dimensional planar Poisson–Voronoi tessellation (PVT, see [6]) induced by an homogeneous Poisson point process  $X_E$  of intensity  $\lambda > 0$ . The PVT has been shown to be a good fit for the street systems of European cities (see [8–10]) and has been widely used to model different urban environments as random tessellations. We will denote by  $S$  the set of edges of  $E$  (representing the streets). The devices are placed on  $S$  as a linear Poisson point process of intensity  $\theta > 0$ , thus forming a Cox point process on the plane with random intensity measure  $\Lambda(B) = \theta|S \cap B|$  for every measurable  $B \in \mathbb{R}^2$ . Here  $|S \cap B|$  stands for the total length of  $S$  in the area  $B$ .

#### 3.1.2 Mobility behavior

Devices move at the same constant speed  $v$  repeating indefinitely the following mobility scheme (random-waypoint model adapted to streets):

- Each device independently picks a destination on the street system. For this we sample a random point  $P$  in the plane using a Gaussian distribution centered on the device  $X$ , and with a standard deviation equal to  $\sigma_X = (15\text{min}) \times v$ . The destination we take for  $X$  is then the closest point of  $P$  in  $E$ . This choice of  $\sigma_X$  shows that devices will go to destinations that they can reach in an average time of 15min if they take a straight path.
- Devices move to their destinations following the shortest path along the streets,
- Once arrived, devices go back to their starting position following again the shortest path along the streets.

#### 3.1.3 Communication behavior

In order to exchange messages, two communicating devices must obey the following rules:

- (RAD): The Euclidean distance between the two devices is less than a given constant threshold  $r$ .
- (LOS): The two devices are on the same street.

The first rule supposes that the emission power of the devices is a constant and that we do not take into account interference. The second rule means that the signal cannot go through the buildings, and that

reflections and diffractions are not taken into account. More precisely, let  $X_{i,t}$  denote the position of device  $X_i$  at time  $t$  and  $\mathcal{N}(X_i, X_j) := \{t \geq 0; |X_{i,t} - X_{j,t}| < r \text{ and } \exists s \in S \text{ such that } (X_{i,t}, X_{j,t}) \in s\}$ . Then  $X_i$  and  $X_j$  are connected at time  $t$  if and only if  $t \in \mathcal{N}(X_i, X_j)$ .

### 3.1.4 Infection behavior

We will follow a standard compartmental model, which is a version of SIR model (see for instance [3]). We will call it SIW (susceptible, infected, white knight). In this model, at time zero, only one device will be in the infected state, while a Cox point process  $X_S$  with intensity  $\theta_S$  will define the susceptible devices, and a Cox point process  $X_W$  with intensity  $\theta_W$ , independent of the former one given the PVT tessellation, will define the white knights. When an infected device is connected to a susceptible device for a time longer than a given threshold  $\rho_I$ , the susceptible device will become infected. When an infected device is connected to a white knight for a time longer than another threshold  $\rho_W$ , the infected device will become a white knight. More precisely, if the device  $X_{i,t}$  is infected and if  $[t, t + \rho_I] \in \mathcal{N}(X_i, X_j)$ , then  $X_{j,t+\rho_I}$  is infected. In the same way, if the device  $X_{i,t}$  is a white knight, and if  $[t, t + \rho_W] \in \mathcal{N}(X_i, X_j)$ , then  $X_{j,t+\rho_W}$  is a white knight.

## 3.2 Mapping the stochastic geometry model to the ABM framework

Let us next describe how the multi-layered stochastic geometry model is represented in the ABM framework. For this, note first that the environment is modeled as an undirected graph, relying on some stochastic geometry concepts, as described in 3.1.1.

As for the ABM description, we define our system of virus propagation in D2D as consisting of a finite number of agents, states, actions and rules

$$\text{MAS} := \langle \mathcal{A}, \text{St}, \text{Act}, \text{R}, \text{T} \rangle.$$

More precisely, we consider a set of  $n$  agents  $\mathcal{A} = \{a_i: i \in [1, n]\}$  and a state space  $\text{St} = \{S, I, W\}$  where  $S$  refers to the agent state *susceptible*,  $I$  to *infected* and  $W$  to *white knight*. Further,  $\text{Act} = \{\text{move}, \text{discover}, \text{connect}, \text{infect}, \text{remove}\}$  denotes the set of *actions* that each agent can perform according to its state/type.  $\text{R}$  represents the set of the *behavioral rule base*. Time  $\text{T}$  is assumed to be divided in time units called *slots*, where each slot  $k$  is a positive integer.

Initially, agents of type *susceptible* and *white knight* are respectively distributed on the edges of  $E$  (i.e., streets of the city) as described in Section 3.1. One agent of type *infected* is introduced around the centre of the map.

Formally, each agent  $a_i$  is defined at each time slot by a tuple

$$\mathcal{M}_{i,k} := \langle X_{i,k}, V_{i,k}, N_{i,k}, \text{Act}_{i,k}, \xi_{i,k}, T_{i,k}^{(I)}, T_{i,k}^{(W)} \rangle.$$

Here  $X_{i,k}$  specifies the agent's *location* in terms of coordinates at time  $kdt$ ,  $V_{i,k} = v$  represents the agent's moving *velocity* and  $N_{i,k}$  represents the *knowledge base*, representing what each agent  $a_i$  knows about its neighborhood agents and the environment at time slot  $k$ .  $\xi_{i,k} \in \text{St}$  represents the state of agent  $a_i$ .  $\text{Act}_{i,k}$  is the set of actions that could be performed by  $a_i$ . Further,  $T_{j,k}^{(I)}, T_{j,k}^{(W)}$  represent respectively the *first time* when  $a_i$  becomes infected and the first time when it becomes a white knight. The variables  $T_{j,k}^{(I)}, T_{j,k}^{(W)}$  will be updated during the simulation depending on the agent's

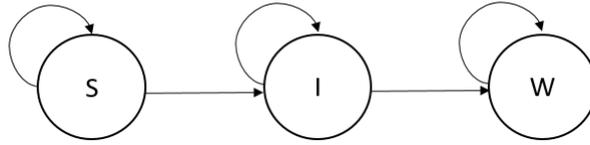


Figure 1: State diagram

interactions. They are both initially set to  $+\infty$  for susceptible agents.  $T_{i,0}^{(I)}$  and  $T_{i,0}^{(W)}$  are respectively set to 0 and  $+\infty$  for infected agents and to  $+\infty$  and 0 for white knight ones. These variables allow to track the state of agents over time. The state  $\xi_{i,t}$  of  $a_i$  at  $t \geq 0$  is either white knight if  $t \geq T_{i,k}^{(W)}$ , infected if  $T_{i,k}^{(I)} \leq t < T_{i,k}^{(W)}$  or susceptible if  $t < T_{i,k}^{(I)}$ .

### 3.3 Agent states

Agent states specify what state an agent is in. Agent state transitions are driven by the rule base  $R$ , see Figure 1 for an illustration. Rules are represented, in this model, by deterministic condition-action rules where each rule consists of a condition part and an action part (like e.g., “if condition1 then action1”). The rule base  $R$  implements the reactive behavior of agents. It allows to select actions to take for agent  $a_i$  depending on its current local state  $\xi_{i,k}$  and its knowledge base  $N_{i,k}$ . More specifically, we write  $R = \{\Theta\}$  where  $\Theta(\xi_{i,k}, N_{i,k})$  are the *active rules*, which map the set of states and observations to actions for reactive tasks

$$\Theta : (\xi_{i,k}, N_{i,k}) \longrightarrow \text{Act}_{i,k}.$$

Let  $T_{i,j}^{(C)}$  be the connection duration between agents  $a_i$  and  $a_j$ ,  $\rho_I$  be the needed time for the virus transmission from one agent to another and  $\rho_W$  the time needed for malware removal. The rule-based functions are described as follows.

- Malware infection rule: If agent  $a_i$  is infected, agent  $a_j$  is susceptible ( $\xi_{i,k} = I, \xi_{j,k} = S$ ) and  $a_i$  was connected to  $a_j$  for a time longer than the infection threshold ( $T_{i,j}^{(C)} \geq \rho_I$ ), then the state of agent  $a_j$  will be transited from susceptible to infected (the action infect will be activated),

$$\Theta_I : (\xi_{i,k}, N_{i,k}) \longrightarrow \text{Infect}.$$

- Malware removal rule: If agent  $a_i$  is a white knight, agent  $a_j$  is infected ( $\xi_{i,k} = W, \xi_{j,k} = I$ ) and  $a_i$  was connected to  $a_j$  for a time longer than the malware removal threshold ( $T_{i,j}^{(C)} \geq \rho_W$ ), then the state of agent  $a_j$  will be transited from infected to white knight (the action remove will be activated).

$$\Theta_W : (\xi_{i,k}, N_{i,k}) \longrightarrow \text{Remove}.$$

A more detailed description of the algorithms associated to malware infection/removal will be given in Section 4.

### 3.4 Agent mobility

In addition to malware infection and removal behavior, managed by the rule base, agents have other actions that are continuously triggered over the simulation: move and discover. More precisely, each

agent will have a constant speed during the simulation and will move according to a modified version of the random-waypoint model, called *A\*Waypoint*, that we propose to reduce computational complexity. Details about this algorithm will be given in Section 4.1. We also will consider that all the agents can move on any street of the map in the two directions, and that the streets do not have any limitation on the maximum number of agents they can contain.

## 4 The multi-agent simulation

In this section we present more details on the implementation of our multi-agent simulation tool. First, we list the key model parameters in Table 1.

Parameter	Description
$dt$	Elapsed time in each step (s)
$\rho_I$	Connection time needed for virus transmissions between agents (s)
$\rho_W$	Connection time needed for antivirus transmissions between agents (s)
$r$	Communication radius of agents (km)
$\lambda$	Intensity of Voronoi seeds (seed/km <sup>2</sup> )
$\theta_S$	Intensity of susceptible agents (agent/km)
$\theta_W$	Intensity of white knights (agent/km)
$v$	Speed of agents (km/h)

Table 1: Simulation parameters

Let

$$\mathcal{P} := \{dt, \rho_I, \rho_W, r, \lambda, \theta_S, \theta_W, v\}$$

denote the set of parameters as presented in Table 1. Other parameters such as the dimensions of the map can be added to this list, but we will not focus on these in our study. Note that we decided to give the same speed to all the agents in order to keep a restraint number of parameters, but we can easily have a more general model where the speeds of the agents are distributed following some probability law. Each agent could have for example a speed taken uniformly at random in some interval  $[v_1, v_2]$ .

Our simulation is done over steps, each step corresponds to a time instant  $kdt$ . In the following we will denote by  $\mathcal{M}_k$  the model at step  $k$ . It represents the map, the agents and all their attributes (coordinates, states, etc.) at step  $k$ .

In the simulation, we first generate a random map, then the agents, and after that we run the function  $\text{Step}(\mathcal{M}_k)$ , that updates the variables of the model, taking it from a step  $k$  to the next step  $k + 1$ , for a number  $k_{\max}$  of iterations. Algorithm 1 describes the entry function of the simulation.

The function  $\text{GenerateMap}(\lambda)$  returns a random PVT based on the parameter  $\lambda$ , whereas the function  $\text{Generate}(\theta_S, \theta_W, v)$  returns the set of agents  $\mathcal{A} := \mathcal{A}_S \cup \mathcal{A}_W \cup \{a_{i_0}\}$ , where  $\mathcal{A}_S$  and  $\mathcal{A}_W$  are respectively the sets of initially susceptible and white-knight agents distributed on  $\mathbb{M}$  using homogeneous Poisson point processes with parameters  $\theta_S$  and  $\theta_W$ , and  $a_{i_0}$  is the initially infected agent, chosen uniformly somewhere near the center of the map.

---

**Algorithm 1:** Main( $\mathcal{P}, k_{\max}$ ): The main function describing the simulation

---

**Input :** The set of parameters  $\mathcal{P}$  and the maximum number of steps  $k_{\max}$

**Output:** The state of a randomly generated model at time  $k_{\max}dt$

```

1  $\mathbb{M} \leftarrow \text{GenerateMap}(\lambda)$ ;
2  $\mathcal{A} \leftarrow \text{GenerateAgents}(\theta_S, \theta_W)$ ;
3  $\mathcal{M}_0 \leftarrow (\mathcal{P}, \mathbb{M}, \mathcal{A}, (X_{i,0})_i, (T_{i,0}^{(I)})_i, (T_{i,0}^{(W)})_i)$ ;
4 for  $k \in \{1, \dots, k_{\max}\}$  do
5    $\mathcal{M}_k \leftarrow \text{Step}(\mathcal{M}_{k-1})$ ;
6 return  $\mathcal{M}_{k_{\max}}$ 

```

---

## 4.1 Mobility algorithm: A\* WayPoint

The classical WayPoint algorithm is a mobility algorithm where each agent has a *home*, that is some point on the map. When an agent is at its home, it chooses a destination on the map, goes towards it via a shortest path, and then comes back to its home. Each agent repeats this process indefinitely.

In our simulation, we consider the homes of the agents and the destinations they choose to be nodes of the map. Whenever an agent  $a_i$  comes back to its home, the choice of a new destination will be done as follows:

- Sample a point  $(x_d, y_d)$  using a Gaussian distribution centered on the home of  $a_i$  and with a standard deviation  $\sigma_D$ ,
- Set the destination of  $a_i$  to the node of the map that is the closest to  $(x_d, y_d)$  in terms of the euclidean distance.

$\sigma_D$  is independent of the agent, but in the scenario where agents can have different speeds, we can imagine replacing  $\sigma_D$  by  $v_i\sigma_T$ , where  $v_i$  is the speed of the agent, and the standard deviation depends linearly on its speed, meaning that the faster is  $a_i$  the further it can go.

The classical WayPoint algorithm has the advantage of delivering a realistic movement of the agents and being easy to implement. The problem in using it lays in the time complexity of finding shortest paths from the homes of the agents to their destinations. That is why we propose a modified version of the WayPoint algorithm inspired by the  $A^*$  algorithm.

Our adaptation consists simply in going to the destination via a "good enough" path instead of a shortest path: a path to the destination can be constructed by greedily choosing the neighboring node that minimizes the Euclidean distance to the destination. That is to say that when an agent is heading towards a direction  $P$ , and is on a node  $Q$  having neighbors  $Q_1, \dots, Q_s$ , the agent will go to node  $Q_i$  such that  $i = \arg \min_{1 \leq j \leq s} d(Q_j, P)$ , where  $d(\cdot, \cdot)$  is the Euclidean distance. Since degrees of the nodes are equal to 3 with probability 1 in PVTs, this decision can be made in a  $O(1)$  time.

However, agents moving with this algorithm can be stuck in cycles. To overcome this, we make each agent memorize each node it visits with a probability  $p_m \ll 1$ , and we do not allow it to go to memorized nodes. In order to prevent some other unwanted phenomena that might occur, when an agent visits a new node, its whole memory is erased with a probability  $p_e \ll p_m$ . This simple solution is efficient and saves memory space. Our experiments gave a mean value of the ratio  $R$  between the lengths of the paths constructed using it and those constructed with the classical WayPoint algorithm that is less than 1.31. We computed this ratio over 100,000 simulations with different values of  $\lambda$ . The values we took for the algorithm's parameters are  $\sigma_D = 1$  km,  $p_m = 0.05$ , and  $p_e = 0.005$ . Note

that one of the main reasons why the  $A^*$  WayPoint is efficient here is the topology of the PVTs: these maps are very strongly connected and a lot of paths exist from each node to another. Our algorithm could turn out to be much less efficient if other types of maps are used.

## 4.2 The step function

Our simulation is done over steps, each step  $k$  corresponds to a time instant  $kdt$ . What we analyze is therefore the state of the system at the discrete instants  $0, dt, 2dt, \dots$ . A difficulty lies in finding a way to correctly update the positions and the states of the agents respecting the virus propagation rules. From step  $k$  to  $k + 1$ , each agent moves independently as described in Section 4.1, which means that in our simulation, we can access the position of each agent at the beginning of each step, and we know also its speed and the edges it has been through, but we do not know if it met other agents, if some of them were infected, or if it connected to one of them for more than  $\rho_I$ . Retrieving this information would be very complicated given only the paths taken by the agents between  $t$  and  $t + dt$ , and it would also require a big time complexity. To overcome this, we first impose the constraint  $dt \leq \min\{\rho_I, \rho_W\}$ . This guarantees that, by only observing the positions of the agents at the time instants  $kdt$  for  $k \in \mathbb{N}$ , we will not miss any two devices that connect for a duration longer than  $\rho_I$  or  $\rho_W$ . Let  $k \in \mathbb{N}$ , and let us assume  $a_i, a_j$  are connected to each other at  $kdt$ , i.e., they are on the same street  $s$  and  $\|X_{i,k} - X_{j,k}\| \leq r$ . We will treat the general case where they can have different speeds  $v_i$  and  $v_j$ , and we will compute the connection duration of  $a_i, a_j$  using their movement equations.

Let us denote by  $t_{i,s}^{(in)}$  (respectively  $t_{i,s}^{(out)}$ ) the time when  $a_i$  gets in (respectively out of) the street  $s$ . These can easily be computed knowing  $X_i$  and the length  $L(s)$  of the street  $s$ . Since  $s$  has two different directions, we need to consider their velocities  $\mathbf{v}_i, \mathbf{v}_j$ . Let  $P_1, P_2$  be the positions of the two extremities of the street  $s$ , let us fix a direction  $\mathbf{e} := (P_2 - P_1)/\|P_2 - P_1\|$  (we can take  $-\mathbf{e}$  as well), and  $\nu_i, \nu_j$  be such that  $\mathbf{v}_i = \nu_i \mathbf{e}, \mathbf{v}_j = \nu_j \mathbf{e}$ . Recall that the absolute speed  $v_i$  of  $a_i$  satisfies  $v_i = \|\mathbf{v}_i\| = \pm \nu_i$  and the same holds for  $a_j$ . Finally, let us also define the coordinates of  $a_i, a_j$  on the street  $s$  by  $d_{i,k} := (X_{i,k} - P_1) \cdot \mathbf{e}$  and  $d_{j,k} := (X_{j,k} - P_1) \cdot \mathbf{e}$ . Then we have the following result.

**Lemma 1.** *If  $a_i, a_j$  are connected at the time instant  $kdt$ , and if  $\nu_i \neq \nu_j$ , then the larger time interval containing  $kdt$  and during which  $a_i, a_j$  are connected is  $[t_{i,j}^{(C,i)}, t_{i,j}^{(C,f)}]$ , where*

$$t_{i,j}^{(C,i)} := \max\left\{kdt - \frac{d_{i,k} - d_{j,k}}{\nu_i - \nu_j} - \frac{r}{|\nu_i - \nu_j|}, t_{i,s}^{(in)}, t_{j,s}^{(in)}\right\},$$

$$t_{i,j}^{(C,f)} := \min\left\{kdt - \frac{d_{i,k} - d_{j,k}}{\nu_i - \nu_j} + \frac{r}{|\nu_i - \nu_j|}, t_{i,s}^{(out)}, t_{j,s}^{(out)}\right\}.$$

If  $\nu_i = \nu_j$ , then

$$t_{i,j}^{(C,i)} = \max\{t_{i,s}^{(in)}, t_{j,s}^{(in)}\} \text{ and } t_{i,j}^{(C,f)} = \min\{t_{i,s}^{(out)}, t_{j,s}^{(out)}\}.$$

The connection duration of  $a_i, a_j$  is then  $T_{i,j}^{(C)} := t_{i,j}^{(C,f)} - t_{i,j}^{(C,i)}$ .

We saw in Section 3.2 that, at each step  $k \geq 1$ , the states of the agents will be determined by the variables  $T_{i,k-1}^{(I)}, T_{i,k-1}^{(W)}$ . We call  $\mathcal{S}_k, \mathcal{I}_k, \mathcal{W}_k$  the sets of susceptible, infected, and white-knight agents. Let  $\text{ConnectionInterval}(a_i, a_j)$  be a function computing  $t_{i,j}^{(C,i)}, t_{i,j}^{(C,f)}$  as in Lemma 1, and let  $\text{GetNeighbors}(a_i)$  be a function returning the set of neighbors of  $a_i$  defined as  $\mathcal{N}_k(a_i) := \{a_j \in \mathcal{A} : \|X_{i,k} - X_{j,k}\| \leq r \text{ and } a_i, a_j \text{ are on the same street}\}$ . Searching the neighbors of all the agents would normally require a  $O(n^2)$  time complexity, but since only agents on a same street can connect to each other, we considerably reduced this complexity by searching neighbors of each agent only

among those that are on the same street. From here, we can write Algorithm 2 that updates the values  $T_{j,k}^{(I)}$  for the neighbors of an infected agent  $a_i$ .

---

**Algorithm 2:** InfectNeighbors( $a_i$ )
 

---

**Input** : An infected agent  $a_i$

**Output:** Updates  $T_{j,k}^{(I)}$  for all susceptible neighbors of  $a_i$

```

1  $\mathcal{N}_{i,k}^{(S)} \leftarrow \text{GetNeighbors}(a_i) \cap \mathcal{S}_k$ ;
2 for  $a_j \in \mathcal{N}_{i,k}^{(S)}$  do
3    $t_{i,j}^{(C,i)}, t_{i,j}^{(C,f)} \leftarrow \text{ConnectionInterval}(a_i, a_j)$ ;
4    $t_1 \leftarrow \max\{t_{i,j}^{(C,i)}, T_{i,k-1}^{(I)}\}$ ;
5    $t_2 \leftarrow \min\{t_{i,j}^{(C,f)}, T_{i,k}^{(W)}, (k+1)dt\}$ ;
6   if  $t_2 - t_1 \geq \rho_I$  then
7      $T_{j,k}^{(I)} \leftarrow \min\{T_{j,k}^{(I)}, t_1 + \rho_I\}$ ;

```

---

Lines 4 and 5 make sure that we only compute the time when the agents are connected and  $a_i$  is infected and that the update is made only at the last step before the neighbor becomes infected. Also, in the line 7, we cannot set the value of  $T_{j,k}^{(I)}$  simply to  $t_{i,j}^{(C,i)} + \rho_I$  because the agent  $a_j$  might be connected to several infected agents, and it will become infected as soon as it stays connected to one of them for longer than  $\rho_I$ . We can define a similar function HealNeighbors( $a_i$ ), see Algorithm 3, that takes as argument a white knight  $a_i$  and updates the values  $T_{j,k}^{(W)}$  of its infected neighbors. Again, for

---

**Algorithm 3:** HealNeighbors( $a_i$ )
 

---

**Input** : A white-knight agent  $a_i$

**Output:** Updates  $T_{j,k}^{(W)}$  for all infected neighbors of  $a_i$

```

1  $\mathcal{N}_{i,k}^{(I)} \leftarrow \text{GetNeighbors}(a_i) \cap \mathcal{I}_k$ ;
2 for  $a_j \in \mathcal{N}_{i,k}^{(I)}$  do
3    $t_{i,j}^{(C,i)}, t_{i,j}^{(C,f)} \leftarrow \text{ConnectionInterval}(a_i, a_j)$ ;
4    $t \leftarrow \max\{t_{i,j}^{(C,i)}, T_{i,k-1}^{(W)}, T_{j,k-1}^{(I)}\}$ ;
5   if  $t_{i,j}^{(C,f)} - t \geq \rho_W$  then
6      $T_{j,k}^{(W)} \leftarrow \min\{T_{j,k}^{(W)}, t + \rho_W\}$ ;

```

---

each infected neighbor  $a_j$  of  $a_i$ , we must make sure to compute their connection time while  $a_i$  is white knight and  $a_j$  is infected.

Finally, we can write the core function of our simulation, that is Algorithm 4.

Note that it is important to update the variables  $(T_{j,k}^{(W)})_j$  before  $(T_{j,k}^{(I)})_j$ , because the connection time computed when calling InfectNeighbors( $a_j$ ) depends also on  $T_{j,k}^{(W)}$ .

### 4.3 Equivalence of discrete and continuous time

We denote by  $\xi_i(t)$  the state of agent  $a_i$  at *continuous time*  $t$  for any  $a_i \in \mathcal{A}$ , i.e., its state according the mathematical malware propagation model, which is defined in continuous time. On the other hand,

**Algorithm 4:** The step function**Input :** The model  $\mathcal{M}_{k-1}$  at step  $k - 1$ **Output:** The model  $\mathcal{M}_k$  at step  $k$ 


---

```

1 for  $a_i \in \mathcal{A}$  do
2    $X_{i,k} \leftarrow \text{Move}(a_i, V_i, X_{i,k-1}, dt)$ ; //Update the positions
3    $T_{i,k}^{(I)}, T_{i,k}^{(W)} \leftarrow T_{i,k-1}^{(I)}, T_{i,k-1}^{(W)}$ ; //Initialization
4  $\mathcal{S}_k, \mathcal{I}_k, \mathcal{W}_k \leftarrow$  The sets of susceptible, infected and white-knight agents;
5 for  $a_i \in \mathcal{W}_k$  do
6    $\text{HealNeighbors}(a_i)$ ; //Update the variables  $T_{j,k}^{(W)}$ 
7 for  $a_i \in \mathcal{I}_k$  do
8    $\text{InfectNeighbors}(a_i)$ ; //Update the variables  $T_{j,k}^{(I)}$ 
9  $\mathcal{M}_k \leftarrow (\mathcal{P}, \mathbb{M}, \mathcal{A}, (X_{i,k})_i, (T_{i,k}^{(I)})_i, (T_{i,k}^{(W)})_i)$ ;
10 return  $\mathcal{M}_k$ ;

```

---

for each  $k \in \mathbb{N}$ , we denote as before by  $\xi_{i,k}$  the state of  $a_i$  at *discrete time*  $kdt$  as described by our ABM. The following theorem states that, for sufficiently small time slots, at the discrete time points, our model is equivalent to its continuous-time version and is thus equivalent.

**Theorem 2.** *If  $dt < \min\{\rho_I, \rho_W\}$ , then we have*

$$\forall a_j \in \mathcal{A}, \forall k \in \mathbb{N}, \quad \xi_{j,k} = \xi_j(kdt).$$

In words, Theorem 2 guarantees that, via discretizing, we do not miss infection events and the introduced time differences do not induce errors in the discretized model. We define the first continuous times when  $a_j \in \mathcal{A}$  is respectively infected and healed as  $\tilde{T}_j^{(I)} := \inf\{t \geq 0 : \xi_j(t) = I\}$  and  $\tilde{T}_j^{(W)} := \inf\{t \geq 0 \mid \xi_j(t) = W\}$ . Regarding our malware propagation rules, we can write

$$\tilde{T}_j^{(I)} = \inf_{a_i \neq a_j} \inf_{t \geq \tilde{T}_i^{(I)}} \{t + \rho : [t, t + \rho] \subset \mathcal{N}(a_i, a_j)\}, \quad (1)$$

$$\tilde{T}_j^{(W)} = \inf_{a_i \neq a_j} \inf_{t \geq \max\{\tilde{T}_j^{(I)}, \tilde{T}_i^{(W)}\}} \{t + \rho_W : \forall h \in [t, t + \rho_W], a_j \in \mathcal{N}_i(h)\}, \quad (2)$$

where  $\mathcal{N}(a_i, a_j)$  is as defined in Section 3.1.3. Let us also denote  $\mathcal{S}_k := \{a_i : kdt < T_{i,k-1}^{(I)}\}$ ,  $\mathcal{I}_k := \{a_i : T_{i,k-1}^{(I)} \leq kdt < T_{i,k-1}^{(W)}\}$  and  $\mathcal{W}_k := \{a_i : T_{i,k-1}^{(I)} \leq kdt < T_{i,k-1}^{(W)}\}$ , and similarly  $\tilde{\mathcal{S}}_k, \tilde{\mathcal{I}}_k$  and  $\tilde{\mathcal{W}}_k$  by replacing  $T_{i,k-1}^{(I)}$  by  $\tilde{T}_i^{(I)}$  and  $T_{i,k-1}^{(W)}$  by  $\tilde{T}_i^{(W)}$ . They correspond respectively to the sets of agents that are susceptible, infected or white knights, in the simulation and in the continuous-time model at time  $kdt$ . Finally, for convenience, let  $T_{i,-1}^{(I)} := T_{i,0}^{(I)}$  and  $T_{i,-1}^{(W)} := T_{i,0}^{(W)}$  for all  $a_i \in \mathcal{A}$ . We have the following lemma.

**Lemma 3.** *If  $dt < \min\{\rho_I, \rho_W\}$ , then for any  $k \in \mathbb{N}$ ,*

$$(\mathcal{B}_k) : \quad \forall a_j \in \mathcal{A}, \quad \begin{cases} \tilde{T}_j^{(I)} \leq T_{j,k-1}^{(I)}, \\ \tilde{T}_j^{(W)} \leq T_{j,k-1}^{(W)}, \\ \tilde{T}_j^{(I)} \leq kdt \implies \tilde{T}_j^{(I)} = T_{j,k-1}^{(I)}, \\ \tilde{T}_j^{(W)} \leq kdt \implies \tilde{T}_j^{(W)} = T_{j,k-1}^{(W)}. \end{cases}$$

The third statement of  $\mathcal{B}_k$  gives correctness of the states for infected agents, the fourth for white knights, while the first and second guarantee that there are no false infected agents or false white knights in the simulation, and therefore correctness of the states of susceptible agents. We will prove this lemma by induction on  $k$ , and once we are done, we can directly deduce Theorem 2 using the following lemma.

**Lemma 4.** *Let  $k \in \mathbb{N}$ . If the assertion  $\mathcal{B}_k$  is true, then we have*

$$\mathcal{S}_k = \tilde{\mathcal{S}}_k, \quad \mathcal{I}_k = \tilde{\mathcal{I}}_k, \quad \mathcal{W}_k = \tilde{\mathcal{W}}_k,$$

and for any  $a_j \in \mathcal{A}$ :

$$a_j \in \mathcal{I}_k \implies T_{j,k-1}^{(I)} = \tilde{T}_j^{(I)}, \quad a_j \in \mathcal{W}_k \implies T_{j,k-1}^{(W)} = \tilde{T}_j^{(W)}.$$

*Proof.* Let  $k \in \mathbb{N}$  and let us assume that  $\mathcal{B}_k$  is true. For any agent  $a_j \in \mathcal{A}$ , we have the following: Since  $T_{j,k-1}^{(I)} \geq \tilde{T}_j^{(I)}$ , then  $a_j \in \tilde{\mathcal{S}}_k \iff kdt < \tilde{T}_j^{(I)} \implies kdt < T_{j,k-1}^{(I)} \iff a_j \in \mathcal{S}_k$ . The second and third statements of  $\mathcal{B}_k$  give that  $a_j \in \tilde{\mathcal{I}}_k \iff \tilde{T}_j^{(I)} \leq kdt < \tilde{T}_j^{(W)} \implies T_{j,k-1}^{(I)} \leq kdt < T_{j,k-1}^{(W)} \iff a_j \in \mathcal{I}_k$ . The fourth statement of  $\mathcal{B}_k$  gives  $a_j \in \tilde{\mathcal{W}}_k \iff kdt \geq \tilde{T}_j^{(W)} \implies kdt \geq T_{j,k-1}^{(W)} \iff a_j \in \mathcal{W}_k$ . This means that we have  $\tilde{\mathcal{S}}_k \subset \mathcal{S}_k$ ,  $\tilde{\mathcal{I}}_k \subset \mathcal{I}_k$  and  $\tilde{\mathcal{W}}_k \subset \mathcal{W}_k$ , but since  $|\mathcal{A}| = |\tilde{\mathcal{S}}_k| + |\tilde{\mathcal{I}}_k| + |\tilde{\mathcal{W}}_k| = |\mathcal{S}_k| + |\mathcal{I}_k| + |\mathcal{W}_k|$ , we deduce that the previous inclusions are in fact set equalities. Now if  $a_j \in \mathcal{I}_k = \tilde{\mathcal{I}}_k$  then necessarily  $\tilde{T}_j^{(I)} \leq kdt$  and therefore  $\tilde{T}_j^{(I)} = T_{j,k-1}^{(I)}$ , and if  $a_j \in \mathcal{W}_k$  then  $\tilde{T}_j^{(W)} \leq kdt$  and  $\tilde{T}_j^{(W)} = T_{j,k-1}^{(W)}$ .  $\square$

From this lemma, we have directly that if  $\mathcal{B}_k$  is true then  $\forall a_j \in \mathcal{A} : \xi_{j,k} = \xi_j(kdt)$ , and therefore if we prove Lemma 3 then Theorem 2 is true. Now, before starting the proof of Lemma 3, observe that we have the following monotonicities during the steps of our algorithm.

**Proposition 5.** *For any agent  $a_j \in \mathcal{A}$ , we have*

- $\tilde{T}_j^{(W)} \geq \tilde{T}_j^{(I)} + \rho_W$ ,
- the sequences  $(T_{j,k}^{(I)})_k$  and  $(T_{j,k}^{(W)})_k$  are non-increasing.

*Proof.* Let  $a_j \in \mathcal{A}$ . If  $a_j$  is initially susceptible or infected, then the first assertion is trivial given Equation 2, and if  $a_j$  is initially a white-knight then it is again true because  $\tilde{T}_j^{(I)} = -\infty$  and  $\tilde{T}_j^{(W)} = 0$ . For the second assertion, let  $a_j \in \mathcal{A}$  and  $k \in \mathbb{N}$ .  $T_{j,k}^{(I)}$  is initialized in the  $k^{\text{th}}$  call of the step function to the value  $T_{j,k-1}^{(I)}$ . The only place in the simulation where  $T_{j,k}^{(I)}$  is updated is in line 7 of Algorithm 2, and the updating formula is  $T_{j,k}^{(I)} \leftarrow \min\{T_{j,k}^{(W)}, t + \rho_W\}$ . Hence each update assigns to  $T_{j,k}^{(I)}$  a value not larger than the previous one it had, the value it will have by the end of the step  $k$  is then smaller than its initialization  $T_{j,k-1}^{(I)}$ . We prove with similar arguments that  $(T_{j,k}^{(W)})_k$  is non-increasing.  $\square$

*Proof of Lemma 3.* Let us assume that  $dt < \min\{\rho_I, \rho_W\}$ . The proof proceeds via induction on  $k$ .

**Initialization:** For  $k = 0$ , we have for any agent  $a_j \in \mathcal{A}$ :

- If  $a_j$  is initially susceptible, then  $T_{j,-1}^{(I)} = T_{j,-1}^{(W)} = +\infty$  and  $\tilde{T}_j^{(I)} \geq \rho_I$ ,  $\tilde{T}_j^{(W)} \geq \rho_I + \rho_W$ ,
- if  $a_j$  is initially infected, then  $T_{j,-1}^{(I)} = 0$ ,  $T_{j,-1}^{(W)} = +\infty$  and  $\tilde{T}_j^{(I)} = 0$ ,  $\tilde{T}_j^{(W)} \geq \rho_W$ ,
- if  $a_j$  is initially a white-knight, then  $T_{j,-1}^{(I)} = -\infty$ ,  $T_{j,-1}^{(W)} = 0$  and  $\tilde{T}_j^{(I)} = -\infty$ ,  $\tilde{T}_j^{(W)} = 0$ .

In any of these three cases, the assertions given in  $\mathcal{B}_0$  concerning  $a_j$  are satisfied, and therefore  $\mathcal{B}_0$  is true.

**Induction:** Now, let  $k \in \mathbb{N}$  and let us assume that  $\mathcal{B}_k$  is true, then we have the following lemmas in order to verify  $\mathcal{B}_{k+1}$ .

**Lemma 6.** For any agent  $a_j \in \mathcal{A}$  we have the implications

$$\begin{aligned} a_j \in \mathcal{I}_k &\implies \tilde{T}_j^{(I)} = T_{j,k-1}^{(I)}, \\ a_j \in \mathcal{W}_k &\implies \tilde{T}_j^{(I)} = T_{j,k-1}^{(I)} \text{ and } \tilde{T}_j^{(W)} = T_{j,k-1}^{(W)}. \end{aligned}$$

Given Lemma 4, these implications are true with  $\tilde{\mathcal{I}}_k$  and  $\tilde{\mathcal{W}}_k$  instead of  $\mathcal{I}_k$  and  $\mathcal{W}_k$ .

*Proof.* Let  $a_j \in \mathcal{A}$ . If  $a_j \in \mathcal{I}_k$  then by Lemma 4 we have  $a_j \in \tilde{\mathcal{I}}_k = \mathcal{I}_k$ , which means that  $\xi_j(kdt) = I$ . This implies that  $\tilde{T}_k^{(I)} \leq kdt$ , and by the induction hypothesis we deduce that  $\tilde{T}_j^{(I)} = T_{j,k-1}^{(I)}$ .

If  $a_j \in \mathcal{W}_k$ , then similarly we deduce that  $a_j \in \tilde{\mathcal{W}}_k$  and that  $\tilde{T}_j^{(W)} \leq t$ , which implies by the induction hypothesis that  $\tilde{T}_j^{(W)} = T_{j,k-1}^{(W)}$ . Moreover, Proposition 5 gives that  $\tilde{T}_j^{(I)} \leq \tilde{T}_j^{(W)} - \rho_I < \tilde{T}_j^{(W)} \leq kdt$  and therefore  $\tilde{T}_j^{(I)} = T_{j,k-1}^{(I)}$  again by the induction hypothesis.  $\square$

**Lemma 7.** For any agent  $a_j \in \mathcal{A}$  we have  $\tilde{T}_j^{(W)} \leq T_{j,k}^{(W)}$ .

*Proof.* Let  $a_j \in \mathcal{A}$ . If  $T_{j,k}^{(W)} = T_{j,k-1}^{(W)}$  then the result is true by the induction hypothesis. Otherwise,  $T_{j,k}^{(W)}$  was updated during the  $k^{\text{th}}$  call of the step function (Algorithm 4). Therefore there is necessarily an agent  $a_i \in \mathcal{W}_k$  for which  $\text{HealNeighbors}(a_i)$  has been called such that  $a_j \in \mathcal{N}_{i,k}^{(I)} := \mathcal{N}_i(kdt) \cap \mathcal{I}_k$  (See Algorithms 4 and 3), and such that  $T_{i,k}^{(W)} = t + \rho_W \leq t_{i,j}^{(C,f)}$  and  $t := \max\{t_{i,j}^{(C,i)}, T_{i,k-1}^{(W)}, T_{j,k-1}^{(I)}\}$ . We have then  $a_j \in \mathcal{N}_{i,k}^{(I)} \subset \mathcal{I}_k$  and  $a_i \in \mathcal{W}_k$ , Lemma 6 gives  $T_{j,k-1}^{(I)} = \tilde{T}_j^{(I)}$  and  $T_{i,k-1}^{(W)} = \tilde{T}_i^{(W)}$  therefore  $t \geq \max\{\tilde{T}_j^{(I)}, \tilde{T}_i^{(I)}\}$ , and since  $[t, t + \rho_W] \subset [t_{i,j}^{(C,i)}, t_{i,j}^{(C,f)}]$  then  $\forall h \in [t, t + \rho_W] : a_j \in \mathcal{N}_i(h)$ , it follows that

$$\begin{aligned} T_{j,k}^{(W)} = t + \rho_W &\geq \inf_{t' \geq \max\{\tilde{T}_j^{(I)}, \tilde{T}_i^{(W)}\}} \{t' + \rho_W \mid \forall h \in [t', t' + \rho_W] : a_j \in \mathcal{N}_i(h)\} \\ &\geq \inf_{a_z \neq a_j} \inf_{t' \geq \max\{\tilde{T}_j^{(I)}, \tilde{T}_z^{(W)}\}} \{t' + \rho_W \mid \forall h \in [t', t' + \rho_W] : a_j \in \mathcal{N}_z(h)\} = \tilde{T}_j^{(W)}, \end{aligned}$$

as desired.  $\square$

**Lemma 8.** For any  $a_j \in \mathcal{A}$ , we have the implication  $\tilde{T}_j^{(W)} \leq (k+1)dt \implies \tilde{T}_j^{(W)} = T_{j,k}^{(W)}$ .

*Proof.* Let  $a_j \in \mathcal{A}$ . If  $\tilde{T}_j^{(W)} \leq kdt$  then by the induction hypothesis and Proposition 5 we have  $\tilde{T}_j^{(W)} = T_{j,k-1}^{(W)} \geq T_{j,k}^{(W)}$ , and with Lemma 7 we deduce that  $\tilde{T}_j^{(W)} = T_{j,k}^{(W)}$ . Otherwise, we have  $kdt < \tilde{T}_j^{(W)} \leq (k+1)dt$ . Let  $\tilde{t} := \tilde{T}_j^{(W)} - \rho_W$ , by Equation (2) we know that there exists an agent  $a_i$  such that

$$\tilde{t} \geq \max\{\tilde{T}_i^{(W)}, \tilde{T}_j^{(I)}\} \quad \text{and} \quad \forall h \in [\tilde{t}, \tilde{t} + \rho_W] : a_j \in \mathcal{N}_i(h). \quad (3)$$

From this and with  $dt < \min\{\rho_I, \rho_W\}$  we have

$$\max\{\tilde{T}_i^{(W)}, \tilde{T}_j^{(I)}\} \leq \tilde{t} := \tilde{T}_j^{(W)} - \rho_W \leq (k+1)dt - \rho_W < kdt < \tilde{T}_j^{(W)} = \tilde{t} + \rho_W. \quad (4)$$

In particular,  $\max\{\tilde{T}_i^{(W)}, \tilde{T}_j^{(I)}\} \leq kdt < \tilde{T}_j^{(W)}$  implies that  $a_j \in \tilde{\mathcal{I}}_k = \mathcal{I}_k$  and  $a_i \in \tilde{\mathcal{W}}_k = \mathcal{W}_k$  (see Lemma 4), and  $kdt \in [\tilde{t}, \tilde{t} + \rho_W]$  together with (3) imply that  $a_j \in \mathcal{N}_{i,k}^{(I)} := \mathcal{N}_i(kdt) \cap \mathcal{I}_k$ . Now,  $a_i$  being in  $\mathcal{W}_k$  ensures that the function `HealNeighbors` will be called with  $a_i$  as input in Algorithm 4, and  $a_j$  being in  $\mathcal{N}_{i,k}^{(I)}$  ensures that  $a_j$  is visited in the "for"loop of `HealNeighbor(a_i)` in Algorithm 3. During this call, a variable  $t$  will be defined as

$$t := \max\{t_{i,j}^{(C,i)}, T_{i,k-1}^{(W)}, T_{j,k-1}^{(I)}\} = \max\{t_{i,j}^{(C,i)}, \tilde{T}_i^{(W)}, \tilde{T}_j^{(I)}\}.$$

The second equality is a consequence of Lemma 6 because  $a_i \in \mathcal{W}_k$  and  $a_j \in \mathcal{I}_k$ .  $T_{j,k}^{(W)}$  will be updated if and only if  $t_{i,j}^{(C,f)} - t \geq \rho_W$  (Line 5 in Algorithm 3). We will prove that this is verified and that the new value of  $T_{j,k}^{(W)}$  is at most  $\tilde{T}_j^{(W)}$ . Indeed, by (3) and (4) we have  $\forall h \in [\tilde{t}, \tilde{t} + \rho_W] : a_j \in \mathcal{N}_i(h)$  and  $kdt \in [\tilde{t}, \tilde{t} + \rho_W]$ . We deduce that  $[\tilde{t}, \tilde{t} + \rho_W] \subset [t_{i,j}^{(C,i)}, t_{i,j}^{(C,f)}]$ , and thus

$$\tilde{t} \geq t_{i,j}^{(C,i)} \quad \text{and} \quad \tilde{t} + \rho \leq t_{i,j}^{(C,f)}.$$

The first inequality combined with the inequality in (3), gives that  $\tilde{t} \geq \max\{t_{i,j}^{(C,i)}, \tilde{T}_i^{(W)}, \tilde{T}_j^{(I)}\} = t$ . Therefore  $t + \rho_W \leq \tilde{t} + \rho_W \leq t_{i,j}^{(C,f)}$ . The "if"condition in Line 5 of Algorithm 3 is then verified, which means that  $T_{j,k}^{(W)}$  is updated as

$$T_{j,k}^{(I)} \leftarrow \min\{T_{j,k}^{(I)}, t_1 + \rho_I\} \leq t_1 + \rho_I \leq \tilde{t} + \rho_I = \tilde{T}_j^{(W)}.$$

It is possible that `InfectNeighbors` will be called after this with some inputs other than  $a_i$  and that  $a_j$  will be visited again, but regarding the updating formula of  $T_{j,k}^{(W)}$ , it can only decrease and therefore by the end of Algorithm 4 we have  $T_{j,k}^{(W)} \leq t + \rho_W \leq \tilde{t} + \rho_W = \tilde{T}_j^{(W)}$ . This, combined with Lemma 7, gives that  $T_{j,k}^{(W)} = \tilde{T}_j^{(W)}$ .  $\square$

**Lemma 9.** For any agent  $a_j \in \mathcal{A}$  we have  $\tilde{T}_j^{(I)} \leq T_{j,k}^{(I)}$ .

*Proof.* Let  $a_j \in \mathcal{A}$ . If  $T_{j,k}^{(I)} = T_{j,k-1}^{(I)}$  then the result is true by the induction hypothesis. Otherwise,  $T_{j,k}^{(I)}$  was updated during the  $k^{\text{th}}$  call of the step function (Algorithm 4). Therefore there is necessarily an agent  $a_i \in \mathcal{I}_k$  for which `InfectNeighbors(a_i)` has been called such that  $a_j \in \mathcal{N}_{i,k}^{(S)} := \mathcal{N}_i(kdt) \cap \mathcal{S}_k$  (See Algorithms 4 and 2), and such that  $T_{i,k}^{(I)} = t_1 + \rho_I \leq t_2$  with  $t_1 := \max\{t_{i,j}^{(C,i)}, T_{i,k-1}^{(I)}\}$  and  $t_2 := \min\{t_{i,j}^{(C,f)}, T_{i,k}^{(W)}, (k+1)dt\}$ . We have directly  $[t_1, t_1 + \rho_I] \subset [t_1, t_2] \subset [t_{i,j}^{(C,i)}, t_{i,j}^{(C,f)}]$  and therefore  $\forall h \in [t_1, t_1 + \rho] : a_i \in \mathcal{N}_i(h)$ . On the other hand, we have that  $\tilde{T}_i^{(W)} \geq t_2$ . In fact, if  $\tilde{T}_i^{(W)} \leq (k+1)dt$  then by Lemma 8:  $\tilde{T}_i^{(W)} = T_{i,k}^{(I)} \geq t_2$ , and in the other case  $\tilde{T}_i^{(W)} > (k+1)dt \geq t_2$ . Also, since  $a_i \in \mathcal{I}_k$ , we have by Lemma 6 that  $\tilde{T}_i^{(I)} = T_{i,k-1}^{(I)} \leq t_1$ . Finally  $t_1$  verifies  $\tilde{T}_i^{(I)} \leq t_1 \leq \tilde{T}_i^{(W)} - \rho_W$  and  $\forall h \in [t_1, t_1 + \rho] : a_i \in \mathcal{N}_i(h)$ . We then have

$$T_{j,k}^{(I)} = t_1 + \rho_I \geq \inf_{a_z \neq a_j} \inf_{\tilde{T}_z^{(I)} \leq t_1 \leq \tilde{T}_z^{(W)} - \rho_I} \{t + \rho_I \mid \forall h \in [t, t + \rho_I] : a_j \in \mathcal{N}_z(h)\} = \tilde{T}_j^{(W)},$$

as desired.  $\square$

**Lemma 10.** For any agent  $a_j \in \mathcal{A}$ , we have the implication  $\tilde{T}_j^{(I)} \leq (k+1)dt \implies \tilde{T}_j^{(I)} = T_{j,k}^{(I)}$ .

*Proof.* Let  $a_j \in \mathcal{A}$ . If  $\tilde{T}_j^{(I)} \leq kdt$  then by the induction hypothesis and by Proposition 5 we have that  $\tilde{T}_j^{(I)} = T_{j,k-1}^{(I)} \geq T_{j,k}^{(I)}$ , and with Lemma 9 we deduce that  $\tilde{T}_j^{(I)} = T_{j,k}^{(I)}$ . Otherwise, we have  $kdt < \tilde{T}_j^{(I)} \leq (k+1)dt$ , and hence  $a_j \in \tilde{\mathcal{S}}_k = \mathcal{S}_k$  (Lemma 4). Let  $\tilde{t} := \tilde{T}_j^{(I)} - \rho_I$ , by Equation (1) we know that there exists an agent  $a_i$  such that

$$\tilde{T}_i^{(I)} \leq \tilde{t} \leq \tilde{T}_i^{(W)} - \rho_I \quad \text{and} \quad \forall h \in [\tilde{t}, \tilde{t} + \rho_I] : a_j \in \mathcal{N}_i(h). \quad (5)$$

From the previous and with  $dt < \min\{\rho_I, \rho_W\}$  we have the inequalities

$$\tilde{T}_i^{(I)} \leq \tilde{t} \leq (k+1)dt - \rho_I < kdt < \tilde{T}_j^{(I)} = \tilde{t} + \rho_I \leq \tilde{T}_i^{(W)}. \quad (6)$$

In particular we deduce that  $\tilde{T}_i^{(I)} < kdt < \tilde{T}_i^{(W)}$ , which means that  $a_i \in \tilde{\mathcal{I}}_k = \mathcal{I}_k$ , and  $kdt \in [\tilde{t}, \tilde{t} + \rho_I]$  therefore  $a_j \in \mathcal{N}_i(kdt)$  by (5), but we have also that  $a_j \in \mathcal{S}_k$  and thus  $a_j \in \mathcal{N}_{i,k}^{(S)} := \mathcal{N}_i(kdt) \cap \mathcal{S}_k$ . These two properties guarantee that during the  $k^{\text{th}}$  call of the step function (see Algorithm 4), the function InfectNeighbors is called with  $a_i$  as input and  $a_j$  is visited in the "for" loop of InfectNeighbors( $a_i$ ) in Algorithm 2. During this call, when  $a_j$  is being visited, we will have

$$t_1 = \max\{t_{i,j}^{(C,i)}, T_{i,k-1}^{(I)}\}, t_2 = \min\{t_{i,j}^{(C,f)}, T_{i,k}^{(W)}, (k+1)dt\}.$$

$T_{j,k}^{(I)}$  will be updated here if and only if  $t_2 - t_1 \geq \rho_I$  (see Line 6 in Algorithm 2). We will prove that this is indeed the case and that the update value is at most  $\tilde{T}_j^{(I)}$ . Hence we have  $[\tilde{t}, \tilde{t} + \rho_I] \subset [t_{i,j}^{(C,i)}, t_{i,j}^{(C,f)}]$  and thus  $\tilde{t} \geq t_{i,j}^{(C,i)}$ , and since  $a_i \in \mathcal{I}_k$ , Lemma 6 implies that  $\tilde{T}_j^{(I)} = T_{j,k-1}^{(I)}$ . Using this with the first inequality in (6), we deduce that

$$\tilde{t} \geq \max\{t_{i,j}^{(C,i)}, \tilde{T}_j^{(I)}\} \geq \max\{t_{i,j}^{(C,i)}, T_{j,k-1}^{(I)}\} = t_1.$$

We deduce also that  $t_1 + \rho_I \leq \tilde{t} + \rho_I \leq t_{i,j}^{(C,f)}$ . What is left is to prove that  $t_1 + \rho \leq \min\{T_{i,k}^{(W)}, (k+1)dt\}$ , for that we will distinguish two cases:

- If  $\tilde{T}_j^{(W)} \leq (k+1)dt$  then, by Lemma 10, we have  $T_{i,k}^{(W)} = \tilde{T}_j^{(W)}$  and thus  $t_1 + \rho_I \leq \tilde{t} + \rho_I \leq \tilde{T}_j^{(W)} = \min\{T_{i,k}^{(W)}, (k+1)dt\}$ .
- In the other case where  $\tilde{T}_j^{(W)} > (k+1)dt$ , we have, by Lemma 9, that  $T_{j,k}^{(I)} \geq \tilde{T}_j^{(I)} > (k+1)dt$ , and finally, with (6), we have  $\tilde{t} + \rho_I \leq (k+1)dt = \min\{T_{i,k}^{(W)}, (k+1)dt\}$ .

Hence we have  $t_1 + \rho_I \leq \min\{t_{i,j}^{(C,f)}, T_{i,k}^{(W)}, (k+1)dt\} = t_2$ . This means that  $T_{j,k}^{(I)}$  will be updated in InfectNeighbors( $a_i$ ) such as

$$T_{j,k}^{(I)} \leftarrow \min\{T_{j,k}^{(I)}, t_1 + \rho_I\} \leq t_1 + \rho_I \leq \tilde{t} + \rho_I = \tilde{T}_j^{(I)}.$$

In the next eventual calls of InfectNeighbors,  $T_{j,k}^{(I)}$  can only decrease and therefore its final value by the end of the step is such that  $T_{j,k}^{(I)} \leq \tilde{T}_j^{(I)}$ . Regarding the result of Lemma 9, this means that  $T_{j,k}^{(I)} = \tilde{T}_j^{(I)}$ .  $\square$

### Conclusion of the proof of Lemma 3

By assuming that  $dt < \min\{\rho_I, \rho_W\}$  and  $\mathcal{B}_k$  is true, we proved Lemmas 7, 8, 9 and 10 proving that  $\mathcal{B}_{k+1}$  is also true. This concludes our proof by induction. We deduce that  $\mathcal{B}_k$  is true for any  $k \in \mathbb{N}$  and therefore, by Lemma 4, we have that:

$$\forall k \in \mathbb{N} : \quad \mathcal{S}_k = \tilde{\mathcal{S}}_k, \quad \mathcal{I}_k = \tilde{\mathcal{I}}_k, \quad \mathcal{W}_k = \tilde{\mathcal{W}}_k,$$

and this means exactly that  $\forall a_j \in \mathcal{A}, \forall k \in \mathbb{N}, \xi_{j,k} = \xi_j(kdt)$ .  $\square$

## 5 Malware propagation in D2D networks: A theoretical study

The model that we presented so far is very rich with many parameters. It is therefore difficult to run simulations varying all these parameters and see how each of them influences the propagation of the virus. So, in order to better choose the values we will assign to them, in this section, we present a theoretical study on a simplified model to identify critical relationships between parameters and values that will lead to drastic changes in the system's evolution. Let us highlight that we consider a different model that does not arise as a limiting object. It is mainly introduced in order to sharpen the intuition for threshold values of important parameters.

The model we consider in this section does not contain white knights, i.e., we only consider susceptible and infected agents. To lighten the equations, we will write simple  $\rho$  instead of  $\rho_I$  and  $\theta$  instead of  $\theta_S$  as there is no confusion.

As in the first model, we start with a single infected agent  $a_{i_0}$ , and we will take interest in the time of the first virus transmission, which we will denote by  $\tau$  in the following. Let us stress that the simplified model that we present here is used only as a mathematical model. All the simulation results in Section 6 are based on the original model and not this simplified one.

We consider the following mean-field approximation of our spatial model. Instead of considering  $a_{i_0}$  to be moving on a PVT, we will consider that it moves on a succession of streets  $s_0, s_1, \dots$ , each having a length  $L_\lambda^{(i)}$  that is a random variable with density  $f_{\lambda,L}$ , where  $f_{\lambda,L}$  is the density function of the edges lengths in a PVT having a seeds intensity equal to  $\lambda$  (see Section 3.1.1). We will assume that, when  $a_{i_0}$  enters a street, other agents are distributed on it as an homogeneous Poisson point process with parameter  $\theta$ , and that they can move in any of the two possible directions. What we mainly lose in this simplified model is the dependence between the lengths of the successive streets visited by  $a_{i_0}$ .

For each street  $s_i$  visited by  $a_{i_0}$ , let  $C_i$  be the number of agents that  $a_{i_0}$  infects while being on  $s_i$ . Let  $p := \Pr[C_i \geq 1]$  denote the probability that  $a_{i_0}$  infects at least some agent on  $s_i$  ( $p$  is independent of  $i$ ). Finally, let  $\tau$  be the first time instant when some agent  $a_j$  different from  $a_{i_0}$  becomes infected

$$\tau := \inf\{t \geq 0 : \exists j \neq i_0 \text{ such that } \xi_j(t) = \text{infected}\}.$$

Then, we have the following main results whose proofs can be found in [1].

**Theorem 11.** *If  $\tau$  is the first time when  $a_{i_0}$  infects another agent, then*

$$\frac{2}{3\sqrt{\lambda v}}(1/p - 1) \leq \mathbb{E}[\tau] \leq \frac{2}{3\sqrt{\lambda v}} \cdot 1/p.$$

**Theorem 12.** *There exists a positive constant  $\tilde{C}$  such that if  $p$  is sufficiently small, then for  $t_0 = 1/(3\sqrt{p\lambda v})$  we have*

$$\Pr[\tau \geq t_0] \geq 1 - \tilde{C}p^{1/4}.$$

These theorems indicate that, if the probability of infecting another agent on a single street  $s_i$  is low, then the waiting time before the virus transmission is very large, and therefore the virus propagation is weak. In terms of the asymptotic behavior of the system, we can state that, when  $p = o(1)$ , then  $\mathbb{E}[\tau] = \Omega(1/(\sqrt{\lambda v p}))$  and for  $t_0 = 1/(3\sqrt{p\lambda v})$  we have  $\Pr[\tau \geq t_0] = 1 - O(p^{1/4})$ .

## 6 Simulation results

This section presents and discusses simulations that were performed in order to analyze malware propagation in D2D. First, we study virus propagation in PVTs without counter measures in order to benchmark the predictions made in 5 and to understand how the different parameters accelerate or slow down the propagation. Then, we study the evolution of the system when white knights are introduced in the network. Conclusions on the conditions of survival or extermination of the virus have been also drawn.

### 6.1 Simulation environment

Our ABM was built based on Mesa [14], which is a very suitable python framework for ABM. We have extended Mesa framework to generate and visualize street system environments. The graphical device interface was implemented using JavaScript and it allows visualizing the simulation evolution over time, tuning the key parameters and visualizing the results to efficiently analyze information.

### 6.2 Evaluation indicators for malware propagation

We present some indicators that allow us to analyze and evaluate the malware propagation. They should be independent of the dimensions of the map, since we theoretically want to study propagation on an infinite plan, hence the interest in considering the propagation speed and the infection rate.

**Definition 13 (Propagation speed).** *The propagation speed of the virus is the speed with which it spreads in space. It is defined by*

$$\mathcal{V} := \limsup_{u \rightarrow +\infty} u\mathbb{E}[1/\tau_u],$$

with  $\tau_u$  the first time when the infection reaches the distance  $u$  from the initial infection point:

$$\tau_u := \inf\{t \geq 0 \mid \exists a_j \in \mathcal{I}(t) : \|X_j(t) - X_{I_0}(0)\| \geq u\}.$$

$\mathcal{I}(t)$  is the set of infected agents at time  $t$ ,  $a_{I_0}$  the only initially infected agent and  $X_{I_0}$  its position at time 0, that is the apparition place of the virus.  $a_{I_0}$  is always chosen close to the center of the map.

**Definition 14 (Infection rate).** *Inspired by Definition 13, we define the infection rate as the rate of infected agents in the region reached by the virus*

$$\mathcal{R} := \limsup_{u \rightarrow +\infty} \frac{|\mathcal{I}(\tau_u)|}{|\{X_j(\tau_u) \mid a_j \in \mathcal{A}\} \cap B(X_{I_0}(0), u)|},$$

where  $B(X_{I_0}(0), u)$  is the open ball of center  $X_{I_0}(0)$  and radius  $u$ , and  $\tau_u$  is as in Definition 13.

Note that  $|\{X_j(\tau_u)\} \cap B(X_{I_0}(0), u)|$  is simply the number of agents inside  $B(X_{I_0}(0), u)$  at time  $\tau_u$ .

$\mathcal{V}$  and  $\mathcal{R}$  are defined as limits, let  $\mathcal{V}_u$  and  $\mathcal{R}_u$  be the expressions in Definitions 13 and 14 that converge to them respectively. To study the behavior of the system, we will only set a value of  $u$  large enough and observe the  $\mathcal{V}_u, \mathcal{R}_u$  considering that they approximate sufficiently the asymptotic values.

In order to avoid boundary effects because the agents would come too close to the boundary of the map, we should have maps of side length  $H$  sufficiently large compared to  $u$ , at least  $H > 2u$ .

### 6.3 Simulation settings

For all simulations, unless otherwise stated, parameters are set by default as follows: ( $u = 3.5\text{km}$ ,  $H = 10\text{km}$ ,  $\lambda = 50\text{km}^{-2}$ ,  $\theta_S = 3\text{km}^{-1}$ ,  $v = 5\text{km/h}$ ,  $\rho_I = 20\text{s}$ ,  $r = 200\text{m}$ ). We assume  $\rho_W = \rho_I$  and  $dt = 0.9 \min\{\rho_I, \rho_W\} = 0.9\rho_I$ . For agent mobility, the  $A^*$  WayPoint algorithm was configured with the parameters  $\sigma_D = 600\text{m}$ ,  $p_m = 0.05$  and  $p_e = 0.005$ , see 4.1. Each value in the diagrams we will present later is the average over 20 simulations with the same set of parameters. In the simulations where  $\lambda$  does not vary, we use the same 20 maps for all the points.

Since we want to observe  $\mathcal{V}_u$  and  $\mathcal{R}_u$ , we will stop the simulations as soon as some agent reaches a distance  $u$  from the initial infection point. To do so, at each step  $k$  of the simulation, we define the maximum propagation radius as  $u_k := \max\{\|X_{j,k} - X_{I_0}(0)\| \mid a_j \in \mathcal{A}\}$ . The simulation ends when  $u_k \geq u$ . In the cases where the virus does not propagate, this condition will never be verified. The reason behind this can be either because  $a_{I_0}$  could not infect other agents, or because the virus was exterminated by the white-knights. In the first case, we will have  $u_k/(kdt) \rightarrow 0$ , and we need to choose a threshold for which we consider the convergence to be achieved, we take it equal to  $0.05\text{km/h}$ . In the second case, we will stop when  $|\mathcal{I}_k| = 0$ . The stopping condition of our simulation is therefore

$$(u_k \geq u \text{ or } u_k/(kdt) < 0.05\text{km/h or } |\mathcal{I}_k| = 0), \quad (7)$$

and the output values are  $\mathcal{V}_{u_k}, \mathcal{R}_{u_k}$ . In the case where  $|\mathcal{I}_k| = 0$ , we will return  $\mathcal{V}_{u_k} = 0$ .

### 6.4 Malware propagation without white knights

#### 6.4.1 The threshold $\sqrt{\lambda\rho v}$

Let us omit the white knights now. The different critical regimes seen in Section 5 are relevant and confirm the intuitive expectations one may have for the propagation of the virus. However, the most remarkable result concerns the regime  $\sqrt{\lambda\rho v} \gg 1$ , because the lower bound found for  $\mathbb{E}[\tau]$  grows with a speed of  $x \mapsto \exp(x^2)/x$  in the quantity  $\sqrt{\lambda\rho v}$ , we can thus expect to observe a rather tight threshold at the level of which the propagation of the virus is no longer possible. To have meaningful results, we will vary the value of  $\lambda$  from 10 to  $200\text{km}^{-2}$  and the speed of the agents from 1 to  $90\text{km/h}$ , and the other parameters will be set as described in Section 6.3. However, when  $\lambda$  is very large, the number of agents  $\mathbb{E}[|\mathcal{A}|] = 2\sqrt{\lambda}H^2\theta_S$  will also be very large since even if it is only proportional to  $\sqrt{\lambda}$ , the multiplicative constant is large. To keep a reasonable number of agents, we used maps with side length  $H_\lambda := 20\lambda^{-1/4}$  for each value of  $\lambda$ , and the stopping propagation radius  $u_\lambda := 0.45 \times H_\lambda$  to have  $H_\lambda > 2u_\lambda$ . This will guarantee that the expected number of agents is  $\mathbb{E}[|\mathcal{A}|] = 2400$  ( $\theta_S = 3$ ), and the side lengths will vary from  $\approx 11.24$  to  $\approx 5.32$ .

We observe in Figures 2 and 3 that the rate of infection and the speed of propagation both cancel out above a certain threshold level line  $\sqrt{\lambda}v = C$  with  $C \approx 3/2$ . This confirms the hypothesis of the exponential lower bound of  $\mathbb{E}[\tau]$ , although it is obtained with a simplified mathematical model. It seems however that this threshold is sharper for  $\mathcal{R}$  than for  $\mathcal{V}$ . The reason why we have such a threshold is that the distribution of the edges lengths in a PVT makes it very rare to have edges much larger than the mean edge length  $\mathbb{E}[L_\lambda] = 2/(3\sqrt{\lambda})$ , and therefore when  $a_{I_0}$  never visits an edge larger than  $\rho v$  the virus cannot propagate, because only agents on a same street can connect to each other.

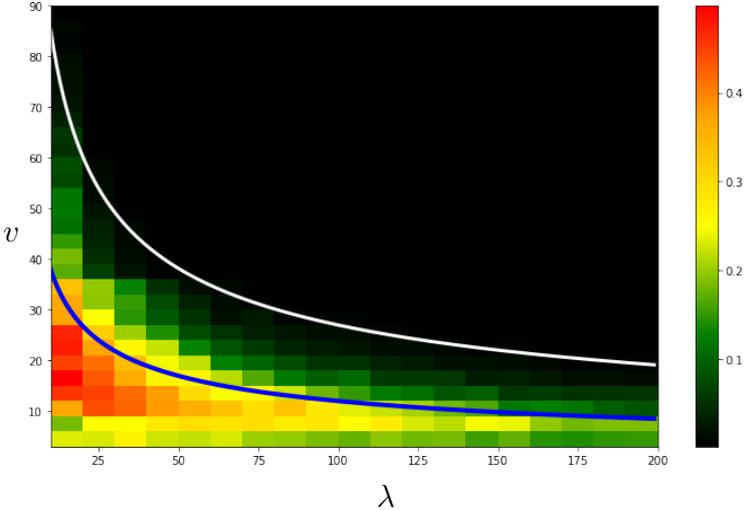


Figure 2: Infection rate  $\mathcal{R}$

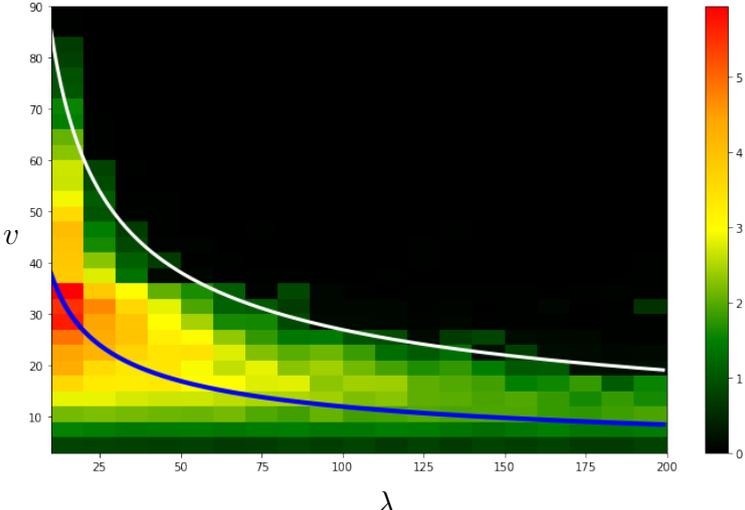
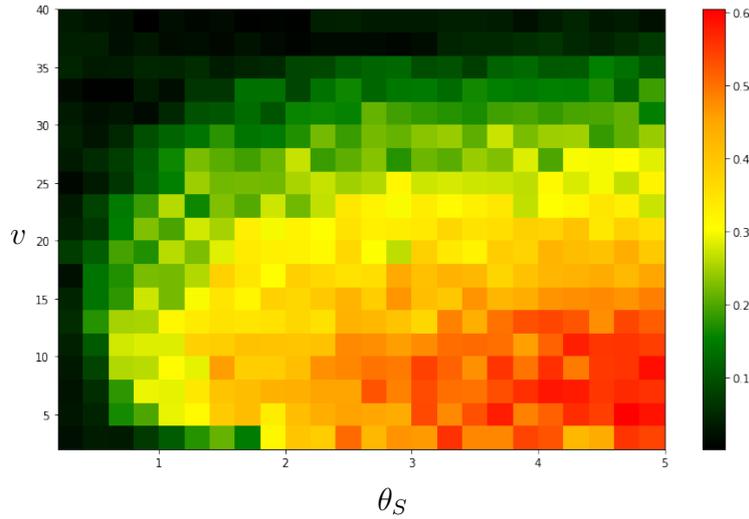


Figure 3: Propagation speed  $\mathcal{V}$  in km/h

Figure 4: Infection rate  $\mathcal{R}$ 

A surprising remark is that the maximum infection rate is always not far below the curve  $\rho v = \mathbb{E}[L_\lambda]$  while the maximum propagation speed seems to be achieved exactly at the points verifying this equation.

In Figure 3 we also observe a lower threshold value of the speed: the virus hardly propagates for  $v = 3$ , but as soon as  $v = 6$ , we see a remarkable jump in the values of  $\mathcal{V}$ . It is to be expected to have a weaker propagation for the small values of the speed because in the limit  $v = 0$  the virus can propagate at most in the street where it was initially placed.

The third observation is that the virus propagation becomes slower as  $\lambda$  becomes larger. The reason is that, as predicted by the simplified model in 5, when  $\sqrt{\lambda}$  becomes much larger than  $\theta_S$ : we have too many streets compared to the number of agents, and therefore  $a_{I_0}$  will only meet a few agents.

#### 6.4.2 How is the propagation speed impacted by $\theta_S$ and $v$ ?

The propagation speed of the virus is certainly a function of all the parameters of our model. However, the distance  $r$  is given by the technology and cannot be changed, and the intensity of streets  $\lambda$  is known for a given city. Now, for a given malware, we want to see the influence of the intensity and speed of users on the propagation speed and the infection rate. In fact, agents that move fast enough but not too fast (to not have  $\sqrt{\lambda}\rho v \geq 3/2$ ) will rapidly carry the virus to the other edges and facilitate its spreading. Also, when the agent's intensity is important, there will always be agents on these streets that will get infected and carry the virus further. Considering Figures 4 and 5, the first thing that stands out, even clearer than in the previous figures, is that the propagation speed and the infection rate show different behaviors. For every  $\theta_S$ , there is clearly an increase and then a decrease of  $\mathcal{V}$  when we increase  $v$ , going from  $\approx 0$  km/h to the maximal value and then returning to 0 km/h. But the value  $\mathcal{R}$  does not change a lot in the first range of values of  $v$ . This means that when agents are slow, they will stay sufficiently long on every street and therefore once an infected agent reaches a street it will infect many agents being on it too, and since  $\mathcal{R}$  takes into account only agents inside the smallest ball containing the virus, it will have important values. Propagation speed is nevertheless slow because the agents take a lot of time before exiting each street and carrying the virus to the next one. This correlation between  $\mathcal{R}$  and  $\mathcal{V}$  confirms the need to study these two quantities together.

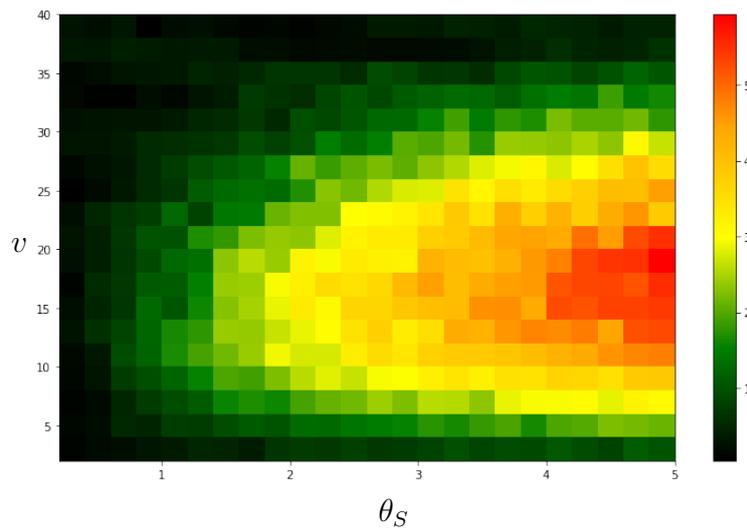


Figure 5: Propagation speed  $\mathcal{V}$  in km/h

Returning to the threshold  $\sqrt{\lambda\rho v}$ , the value of  $v$  verifying  $\sqrt{\lambda\rho v} = 2/3$  is  $v_0 \approx 16.97$ . Thus, we have again that  $\mathcal{R}$  is maximal in the region below the level line  $\sqrt{\lambda\rho v} = 2/3$ , and  $\mathcal{V}$  is maximal exactly in its close neighborhood. This property would therefore be true even when varying  $\theta_S$ . For larger values of  $v$ , we expect that the virus will not propagate anymore because of the streets being too short, and for larger  $\theta_S$ , the propagation can only be more important because the more agents we have the more easily the virus spreads, the propagation speed and the infection rate would however be bounded and will not increase to infinity.

## 6.5 White-knights counter measure

Let us bring back the white knights to our model. Now we will talk not only about the propagation speed and the infection rate, but also about the survival of the virus. We will consider in the simulations that the virus survives if, by the end of the simulation, we have at least 1% of the population that is infected, i.e.,  $\mathcal{R} \geq 0.01$ . Define the virus survival probability by  $\Sigma_{1\%}$ .

Two practical questions we can ask are how many white knights do we need to exterminate the virus, and then how does  $\rho_I$  impact the rate of white knights needed? The answer to the first question is given in Figure 6. It shows that the virus is exterminated beginning from  $\theta_W \geq \theta_S/3$ . Note that this threshold depends on the survival criterion we are considering. If we say for example that the virus survives if more than 10% of the population is infected then we would have different results.

Note that the value of  $\Sigma_{1\%}$  is 1 for the point  $(\theta_S, \theta_W) = (0.1, 0.1)$ . Of course, this cannot be the case. This particular value is biased because of our definition of  $\Sigma_{1\%}$ : for this point the virus does not propagate at all, but neither is it exterminated because there are too few agents and they rarely meet one another. Also because of the small number of agents, we will have that one agent represents more than 1% of the population, and therefore the virus is considered to have survived.

Figure 6 also suggests that only the ratio  $\theta_W/\theta_S$  is important when it comes to the virus survival. This would mean that with a given configuration of the parameters other than  $\theta_S, \theta_W$ , there exists a threshold  $\beta$  such that if  $\theta_W/\theta_S > \beta$  then the virus is eliminated, independently from the values of  $\theta_S$  and  $\theta_W$  themselves. In order to verify this hypothesis, we ran simulations similar to the one in Figure

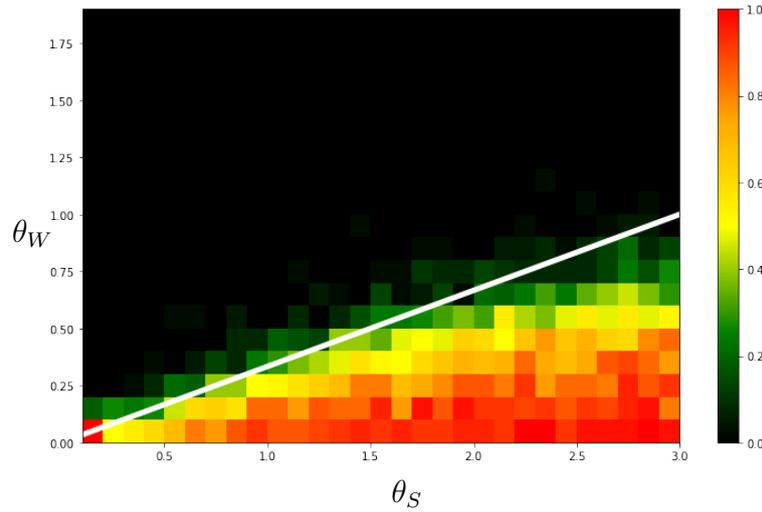


Figure 6:  $\Sigma_{1\%}$  for  $\theta_S \in [0.1, 3](\text{km}^{-1})$ ,  $\theta_W \in [0, 1.9](\text{km}^{-1})$ . In white: the level line  $\theta_W = \theta_S/3$

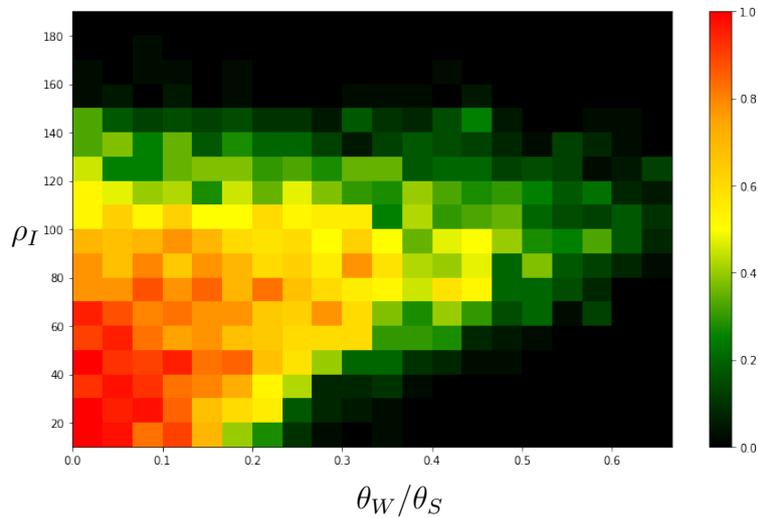


Figure 7:  $\Sigma_{1\%}$  for  $\theta_W/\theta_S \in [0, 2/3]$  and  $\rho_I = \rho_W \in [10, 190]$ .

6 but with different values of  $\rho_I = \rho_W \in \{40, 60, 80\}$ , and we always have similar observations: the virus does not propagate beginning from a threshold value of  $\theta_W/\theta_S$ , different for each  $\rho_I$ . This brings us to answering our second question, that is to identify the dependency between this threshold and  $\rho_I$ , and this is given in Figure 7. We remind that  $\rho_W$  is always taken equal to  $\rho_I$ . The figure was realized by fixing  $\theta_S = 3\text{km}^{-1}$  and varying  $\theta_W$ .

We first observe that in the region where both  $\theta_W$  and  $\rho_I$  are small, the survival is maximum, which is to be expected because the virus is easily transmitted and the white-knights are rare. Then, when  $\rho_I$  increases, through the observations made for Figure 2, the virus will not be able to propagate for  $\rho_I \geq \rho_0$  such as  $\sqrt{\lambda}\rho_0 v = 3/2$ . Numerically we find  $\rho_0 \approx 152.73\text{s}$ , and this corresponds well to what we have in Figure 4.

The relevant phenomenon we observe here is that for  $\rho_I < \rho_0$ , the intensity of white knights needed to eliminate the virus grows with  $\rho_I$ . Moreover, the boundary delimiting the virus survival zone is very thin for small values of  $\rho_I$ , and it becomes more and more spread out when  $\rho_I$  grows. We believe that the reason for this shape of the survival zone is the time lag between the wave of virus propagation

and the wave of white knights propagation. Indeed, the virus will be able to propagate easily at the beginning if the intensity of the susceptible agents is sufficient. On the other hand, white knights will only spread rapidly after there are enough infected agents. This delay between the two waves gives to the virus a step ahead, letting it infect a lot of agents, and hence allowing him a better chance to reach the radius  $u$ .

With this, given the value of  $\theta_S$  and  $\rho_I = \rho_W$ , we can know the white knight's intensity needed for eliminating the virus simply by choosing the suitable ratio  $\theta_W/\theta_S$  from Figure 7.

## Summary of simulation results

From all the simulations we made, we can deduce the following properties:

- The virus does not propagate when  $\sqrt{\lambda\rho_I v} > 3/2$ .
- If we vary,  $\lambda$ ,  $\rho_I$  or  $v$  we find that  $\mathcal{V}$  is maximal on the level line  $\sqrt{\lambda\rho_I v} \approx 2/3$  and  $\mathcal{R}$  is maximal in its neighborhood where  $\sqrt{\lambda\rho_I v} < 2/3$ .
- If  $\rho_I = \rho_W$ , then the intensity of white knights necessary for eliminating the virus would depend on (but not only on) the ratio  $\theta_W/\theta_S$  and  $\rho_I$ . For small values of  $\rho_I$  there is a sharp threshold value of  $\theta_W/\theta_S$  separating the survival and elimination regions of the virus.

## 7 Conclusion and future work

This paper presents a novel agent based simulation model for analyzing malware propagation dynamics in D2D networks. This approach, traditionally applied for complex systems, allows us to obtain relevant and surprising findings about malware propagation in D2D, which demonstrates also its effectiveness for such dynamic communication networks. Notably, malware propagation was not possible above a first threshold ( $\sqrt{\lambda\rho_I v} > 3/2$ ) and was maximal around a second threshold ( $\sqrt{\lambda\rho_I v} = 2/3$ ), which corresponds to having an average length of streets equal to the distance traveled by an agent during the time  $\rho_I$  (needed for infection transmission). This shows, therefore, the importance of street system characteristics, which has been traditionally neglected when studying malware propagation in D2D.

Thanks to this model, two regimes have also been identified using phase diagrams showing boundaries between survival and elimination regions of the virus that depend on devices density (susceptible and white knight) and on the connection time needed for malware infection. An other important result was to estimate the percentage of counter measures (white-knight devices) to introduce in the network in order to eliminate the malware and control its propagation.

We believe agent-based modeling and simulation has a great potential for studying malware spread in D2D communication networks. As a future work, we aim to perform the same simulations on different real urban maps. Comparing results obtained with real street systems and results with probabilistic maps (PVTs), will allow us to deduce to what extent theoretical models can help to predict reality. It would be also possible to study other theoretical models of maps, and try to have a match between maps of real cities and the model that would best fit them.

## References

- [1] Z. Benomar, C. Ghribi, E. Cali, A. Hinsen, and B. Jahnel. Agent-based modeling and simulation for malware spreading in D2D networks. *AAMAS '22*, page 91–99, Richland, SC, 2022.
- [2] T. Blakely, J. Thompson, L. Bablani, P. Andersen, D.A. Ouakrim, N. Carvalho, P. Abraham, M.-A. Boujaoude, A. Katar, and E. Akpan. Determining the optimal CovidFWiopt-19 policy response using agent-based modelling linked to health and cost modelling: Case study for Victoria, Australia. *medRxiv*, 2021.
- [3] F. Brauer, C. Castillo-Chavez, and Z. Feng. *Mathematical models in epidemiology*, volume 32. Springer, 2019.
- [4] M.M. Butt, I. Dey, M. Dzaferagic, M. Murphy, N. Kaminski, and N. Marchetti. Agent-based modeling for distributed decision support in an IoT network. *IEEE Internet of Things Journal*, 7(8):6919–6931, 2020.
- [5] E. Cali, N.N. Gafur, C. Hirsch, B. Jahnel, T. En-Najjary, and R.I.A. Patterson. Percolation for D2D networks on street systems. In *2018 16th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks*, pages 1–6, 2018.
- [6] S.N. Chiu, D. Stoyan, W.S. Kendall, and J. Mecke. *Stochastic Geometry and Its Applications*. Wiley Series in Probability and Statistics. Wiley, 2013.
- [7] A. Drogoul, P. Taillandier, B. Gaudou, M. Choisy, K. Chapuis, N.H. Quang, D. Nguyen-Ngoc, D. Philippon, A. Brugière, and P. Larmande. Designing social simulation to (seriously) support decision-making: Comokit, an agent-based modelling toolkit to analyse and compare the impacts of public health interventions against Covid-19. *Review of Artificial Societies and Social Simulation-RofASSS*, 2020.
- [8] C. Gloaguen and E. Cali. Cost estimation of a fixed network deployment over an urban territory. *Annals of Telecommunications*, nov 2017.
- [9] C. Gloaguen, F. Fleischer, H. Schmidt, and V. Schmidt. Fitting of stochastic telecommunication network models via distance measures and monte-carlo tests. *Telecommunication Systems*, 31:353–377, 2006.
- [10] C. Gloaguen, F. Voss, and V. Schmidt. Parametric distance distributions for fixed access network analysis and planning. In *2009 21st International Teletraffic Congress*, pages 1–8. IEEE, 2009.
- [11] A. Hinsen, B. Jahnel, E. Cali, and J.-P. Wary. Malware propagation in urban D2D networks. In *2020 18th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks*, pages 1–9, 2020.
- [12] A. Hinsen, B. Jahnel, E. Cali, and J.-P. Wary. Phase transitions for chase-escape models on Poisson–Gilbert graphs. *Electronic Communications in Probability*, 25:1 – 14, 2020.
- [13] N.J. Kaminski, M. Murphy, and N. Marchetti. Agent-based modeling of an IoT network. In *2016 IEEE International Symposium on Systems Engineering*, pages 1–7, 2016.
- [14] J. Kazil, D. Masad, and A. Crooks. Utilizing Python for agent-based modeling: The Mesa framework. In Robert Thomson, Halil Bisgin, Christopher Dancy, Ayaz Hyder, and Muhammad Hussain, editors, *Social, Cultural, and Behavioral Modeling*, pages 308–317, Cham, 2020.

- [15] Q. le Gall. *Crowd-networking: Modelling Device-to-Device connectivity on street systems using percolation theory, and economic consequences*. PhD thesis, Université Paris sciences et lettres, 2020.
- [16] Q. le Gall, B. Błaszczyszyn, E. Cali, and T. En-Najjary. The influence of canyon shadowing on device-to-device connectivity in urban scenario. In *2019 IEEE Wireless Communications and Networking Conference*, pages 1–7, 2019.
- [17] M. Maziarz and M. Zach. Agent-based modelling for Sars-CoV-2 epidemic prediction and intervention assessment: A methodological appraisal. *Journal of Evaluation in Clinical Practice*, 26(5):1352–1360, 2020.
- [18] S. Müller, M. Balmer, W. Charlton, R. Ewert, A. Neumann, C. Rakow, T. Schlenther, and K. Nagel. Predicting the effects of Covid-19 related interventions in urban settings by combining activity-based modelling, agent-based simulation, and mobile phone data. *medRxiv*, 2021.
- [19] M. Niazi and A. Hussain. Agent-based tools for modeling and simulation of self-organization in peer-to-peer, ad hoc, and other complex networks. *IEEE Communications Magazine*, 47(3):166–173, 2009.
- [20] M. Niazi and A. Hussain. A novel agent-based simulation framework for sensing in complex adaptive environments. *IEEE Sensors Journal*, 11(2):404–412, 2011.
- [21] M. Pérez-Hernández, B. Alturki, and S. Reiff-Marganiec. FABIOT: A flexible agent-based simulation model for IoT environments. In *2018 IEEE International Conference on Internet of Things and IEEE Green Computing and Communications and IEEE Cyber, Physical and Social Computing and IEEE Smart Data*, pages 66–73, 2018.
- [22] L. Zhang, L. Song, and J. Xu. Preventing malware propagation in D2D offloading networks with strategic mobile users. In *2019 IEEE Global Communications Conference (GLOBECOM)*, pages 1–6. IEEE, 2019.
- [23] L. Zhang and J. Xu. Differential security game in heterogeneous device-to-device offloading network under epidemic risks. *IEEE Transactions on Network Science and Engineering*, 7(3):1852–1861, 2020.