Ad Hoc Networks 50 (2016) 115–127

Contents lists available at ScienceDirect

Ad Hoc Networks

journal homepage: www.elsevier.com/locate/adhoc

MeDrone: On the use of a medical drone to heal a sensor network infected by a malicious epidemic

Nicola Roberto Zema[a,*], Enrico Natalizio[a], Giuseppe Ruggeri[b], Michael Poss[c], Antonella Molinaro[b]

[a] Lab. Heudiasyc, UMR CNRS 7253, France
[b] ARTS Laboratory, University "Mediterranea" of Reggio Calabria - DIIES Department, Italy
[c] LIRMM, UMR CNRS 5506, France

ARTICLE INFO

Article history:
Received 20 January 2016
Revised 16 May 2016
Accepted 17 June 2016
Available online 21 June 2016

Keywords:
Networks of autonomous agents
Optimization
Sensor networks

ABSTRACT

The complexity increase in the software and hardware necessary to support more and more advanced applications for Wireless Sensor Networks conspicuously contribute to render them susceptible to security attacks. The nodes of most complex WSN applications sport desktop-level operating systems and this reliance on software make them ideal prey for traditional threats, like viruses and general malware. To address these problems, in this paper we devise a system for a dedicated mobile node to locate, track, access and cure the infected elements of a WSN, threatened by a proximity malware infection. In parallel, we provide a mathematical formulation for the aforementioned operations. We perform extended simulations, comparing our proposal against classic solutions in different network scenarios and we use the results of the mathematical formulation as a benchmark. Furthermore, we introduce a variation of our proposal, capable to support the concurrent operation of multiple mobile nodes and implement co-operation.

1. Introduction

Proximity malware consists in the injection of malicious code in a network and has become recently a serious issue in Wireless Sensor Networks [1–3,49,50]. The nodes of a WSN are devised traditionally as resource-constrained devices with hard-wired algorithms driving their behavior. Normally, this assumption has made them resilient to the modifications related to the diffusion of software malwares. However, some recent advanced sensing applications [4] use complex sensor nodes that require capabilities comparable to smartphones. Furthermore, as industrial applications expand, devices are developed from commercial, off-the-shelf equipment, which increases the risk of security vulnerabilities [5]. These capabilities usually translate into complex operating systems and communication protocols for the devices and networks operations. For instance, wireless systems for video surveillance are subject to the usual constraints of WSNs, such as the deployment in a difficult to access environment or energy limitations, but some additional complexity is needed to handle a streamed video and transmit it to a remote sink. Furthermore, the deployment of networks tailored for security that employ programmable devices, further increases the chances of security threats. The feasibility of full-blown attacks on WSNs has been studied and, given the relative node isolation of these networks, the majority of threats come from proximity malwares [6]. This kind of malware spreads from one device to another using short-range communication interfaces like Bluetooth and WiFi. Examples of its virulence and dangerousness are available in literature, like the CABIR worm for Bluetooth [7] and the iKee for WiFi suites [8]. Even if several defense mechanisms have been proposed [9,10], so far they rely on external network infrastructure or direct human intervention. They can, for instance, apply patches, isolate malicious nodes, run antiviruses or securing remote accesses. The aforementioned possibilities require direct node access but it is almost completely negated in a remote WSN deployment [11–13] and so different approaches and methods are needed to manage security risks. In this paper we aim at fulfilling this lack of solutions for WSNs and in the following list we explain our contributions:

1. We devise a counter-epidemic protocol that heals the network driving a mobile flying node (referred to as *searcher* in the rest of this paper) through the WSN deployment area after an

---

* Corresponding author.

E-mail addresses: nicola.zema@hds.utc.fr (N.R. Zema), enrico.natalizio@hds.utc.fr (E. Natalizio), giuseppe.ruggeri@unirc.it (G. Ruggeri), michael.poss@lirmm.fr (M. Poss), antonella.molinaro@unirc.it (A. Molinaro).

http://dx.doi.org/10.1016/j.adhoc.2016.06.008

in-network analysis of the malware spreading. The *searcher* is capable of autonomous operations and movement.

2. We propose both a heuristic and an optimization model for the *Targeted Curing Problem* (TCP). The solutions aim at respectively determining the sub-optimal and the best trajectory the *searcher* has to travel to both contain the malware diffusion and cure the nodes, minimizing the time they stay infected. The metrics of diffusion containment are the lowest delay to restore all nodes and the minimization of the number of infected nodes.

In order to show the effectiveness of our findings we use the model output as a benchmark and then we validate our heuristic by comparing it with other movement schemes suited for area coverage.

A preliminary contribution on this topic has been provided by authors in [14]. This paper represents a step forward with respect to [14]: (i) by offering a formal proof that the analytical solution of the TCP in NP hard, (ii) by extending the performance evaluation to include a wider set of scenarios and competitors, and (iii) by providing an outlook on the simultaneous deployment of more than one *searcher*.

This paper is organized as follows: Section 2 introduces the background research on the various subjects this paper refers to, followed, in Section 3, by the set of assumptions we used. The main contribution is described in Sections 4 and 5. In the former we describe the scenario, the main proposal and the mathematical model. In the latter we describe our heuristic-based approach to the same problem and the framework needed to support it. In Section 6 we show the simulation and numerical comparative performance as well as we present some results for the coordination problem with multiple *searchers*. Section 8 ends the paper.

2. Related work

The definition of strategies maximizing the movement efficiency of a tracking entity against an adversary has traditionally been addressed under the area of the *search-and-pursuit* research [15]. According to this topic, a combination of *tracker* entities try to define the best trajectories in respect to an *evader* entity in order to, at the same time, occupy the same space and *capture* it. Considering a malware that spreads among nodes as the aforementioned *evader*, and the *searcher* node as the *tracker*, different methodologies can be applied in order to extract the set of the best trajectories to reach not only the actual infected nodes but also anticipate the spreading and block the epidemic. However, the solutions already available focus on the tracker motion plan analysis to bound its movement performances [16,17]. In this paper we define an actual movement optimization strategy, coupled with a heuristic. Anyway for the peculiar environment our system is put into, its class of dynamics has not been yet treated.

The research on immunization and networks epidemics, which can also be seen as a site percolation problem [18], has been in general a popular topic [19–21]. With reference to WSNs [9,22] it has recently caught the interest from the networking research community. According to the research, a useful strategy limiting the effects of an epidemic is to immunize a sufficiently large subset of the susceptible population [23]. However, according to [24], a computer epidemic is more viral than its real-world counterparts as it is engineered to spread fast. It thus needs almost the 100% of the node population to be immunized to limit the diffusion. The intolerable delays necessary to immunize all the nodes in a WSN make this solution impractical [24]. According to [25], in WSNs, it has been found that the most effective way to restore a node from any condition is to wipe and reload the non-protected (userland, data) part of its operating system and this solution is the one we devised to be implemented by the *searcher*.

More than often, in WSNs and ad-hoc networks, there is a strict bond between the detection of node failures and the detection of outbreaks [1,26] but, as described in [27,28] a malware could be also programmed to not disable the nodes during an *incubation* phase in order to maximize the damage once all the network is infected. As a final note, even if the research on models of malware diffusion on social and proximity network is widespread [29,30], in this work we focus on a curing trajectory setting upon an isolated and difficult to access wireless sensor network rather to model the actual diffusion of a multi-environmental malware.

3. Main assumptions and description

In this Section we describe the main assumptions on which our work is based, and we give some details regarding the epidemic dynamics and the network model. We suppose that sensors have an operating system, computational and communication resources that make them capable to perform advanced applications [4]. Considering the WSN deployment in harsh and remote WSNs scenarios, we have devised to use a flying robot to distribute the cure. Furthermore, we assume that:

1. in the remote WSN installation the only mean of information exchange is over short-range radio links and there is no infrastructured access to an external network;
2. after a variable time a node has been in contact with an infected neighbor, it will become infected itself with the same malware;
3. The identification of malware outbreaks in different kinds of networks has been widely studied and given the various implementations of Intrusion Detection Systems (IDS) [26,31–34] available, we assume that an IDS is active in the network.
4. It is supposed that the geographical localization of nodes is available at least in form of predefined coordinates entered by a human operator when a node is deployed;
5. a node can be cured/immunized by uploading a new image of the operating system;
6. all the messages among nodes, and between nodes and *searcher*s are directly embedded in layer-2 frames, as the primitive used for their exchange are usually embedded in the physical network devices and thus more robust to attacks;
7. we suppose the presence of a low-level secured interface [35] to the nodes operating system as previously described and this feature concretizes in a secure computing interface that is easily to provide as stated in [26].

   Various works [36,37] present the possibility of a secure boot of a node after the retrieval of a new part of the operating system as specified by the guidelines of Trusted Computing [38]. In this way we can assume that nodes could replace parts, modules and possibly their whole operating system by retrieving the necessary software by the means of their wireless interface.

4. Mathematical formulation

In this paper, we introduce the *Targeted Curing Problem* (TCP) as the determination of the positions that a *searcher* should travel to, within an infected WSN, in order to diffuse the *cure*, to limit the malware spreading and to clear the infected nodes. In this Section, we address the formulation of the TPC problem that, for a given network, is capable of determining the optimal set of positions in time and space for the *searcher* to minimize the healing time.

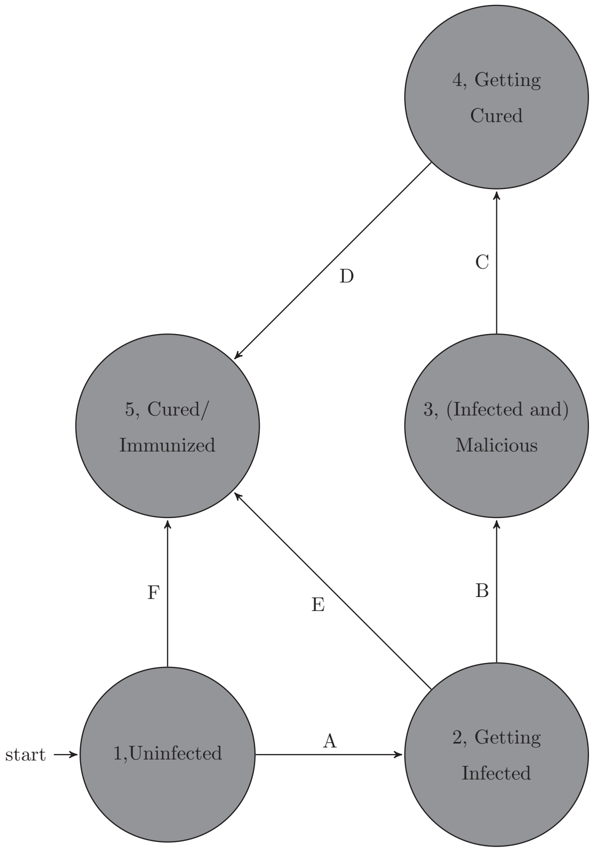

Fig. 1. Nodes states.

Table 1
State transitions.

Transition	States	Happens when
A	1⇒2	A node in state 1 will switch to state 2 if it has been connected for T_a time units to any number of nodes in state 3
B	2⇒3	A node in state 2 will switch to state 3 if any number of nodes in state 3 have been connected to it for, at least, T_t time units
C	3⇒4	A node in state 3 will switch to state 4 if it has been connected for T_a time units to the *searcher*
D	4⇒5	A node in state 4 will switch to state 5 if any number of nodes in state 1 or 2 or the *searcher* are connected to it for, at least, T_t time units
E	2⇒5	A node in state 2 will switch to state 5 if it has been connected for T_a time units to the *searcher*
F	1⇒5	A node in state 1 will switch to state 5 if it has been connected for T_a time units to the *searcher*

In our model, each node can be in only one of five different states at a time. They are illustrated in the diagram of Fig. 1 and are:

1. Uninfected
2. In Transition to Infected
3. Malicious (and Infected)
4. In Transition to Cured
5. Cured/Immunized

The transitions between each couple of states are described in Table 1. Unfortunately, the TCP problem is $\mathcal{NP}$-hard, because it generalizes the problem of looking for a minimum dominating set.

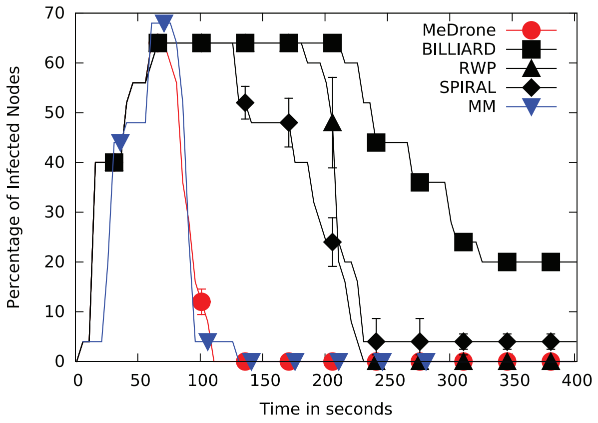

Fig. 2. Percentage of infected elements for the 25-nodes scenario. *Gaussian* Topology.

4.1. $\mathcal{NP}$-hardness for the TCP

1. The TCP problem is $\mathcal{NP}$-hard.

Proof. of $\mathcal{NP}$-hardness for the TCP. Given a graph

$$G = (V, A)$$

and an integer K, the ***DOMINATING SET PROBLEM*** [39] tests whether it is possible to find a subset W of V that contains at most K nodes and such that every other node of V belongs to an edge adjacent to W. Given an instance to the ***DOMINATING SET PROBLEM***, it is defined a corresponding instance for the TCP as follows:

$N = V \cup \{i^*\}$	The set of nodes
$E = A$	The set of wireless links
$E_c \supseteq$ all pairs of nodes in N	The set of physical paths
$N^* = V$	The set of initially infected nodes
$d_e = 1 \; \forall e \in E_c$	Edge lengths
$T_m = T_t = T_a = 0$	The time intervals extracted from the simulated results
$d_i = 0 \; \forall i \in N$	The cure travel times

Then, it can be seen immediately that the answer to the ***DOMINATING SET*** instance is YES if and only if the optimal solution cost to the TCP instance is less than K. □

Hence, no efficient combinatorial algorithm exists for the problem and it is reasonable to address it by using binary integer linear programming. The detailed formulation of the problem in terms of binary variables and linear constraints is given in Appendix A. Unfortunately, the numerical solution determined by using binary integer linear programming is not directly applicable in real cases, because it requires the complete *a-priori* knowledge of the diffusion dynamics of nodes and epidemics. For this reason in Section 5 we developed a heuristic and we retained the integer linear programming results as an upper bound on the quality of the found solution. The results of the comparison, also with other movement schemes, are shown in Section 6.

5. Distributed curing algorithm

The mathematical approach previously described produces a set of waypoints for the *searcher* that, if followed, allows containing the infection and to minimize the malware spreading through the network. Unfortunately, it is required from the model to know precisely the whole infection dynamic of the malware in terms of which node is going to be infected at what time. In a real WSN deployment this last assumption is not feasible and hence we devise

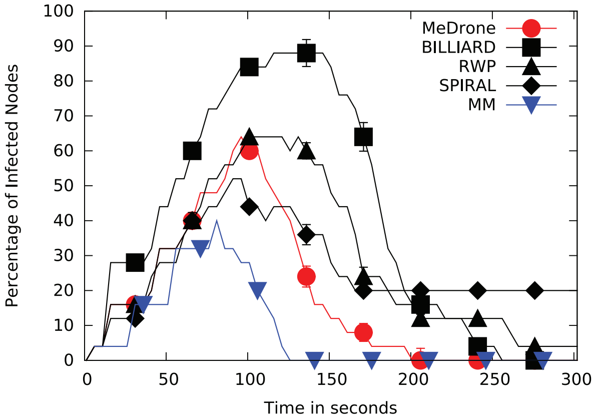

Fig. 3. Percentage of infected elements for the 25-nodes scenario. *Grid* Topology.

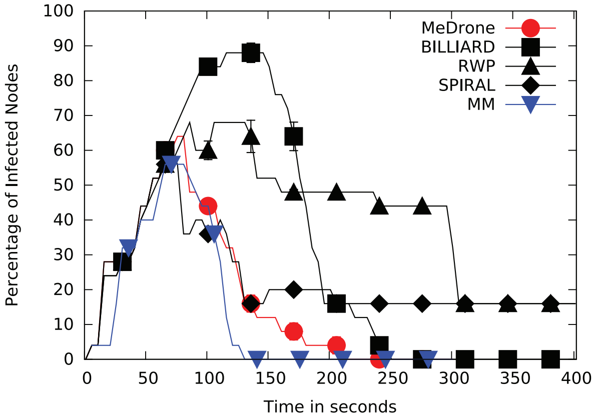

Fig. 4. Percentage of infected elements for the 25-nodes scenario. *Realistc* Topology.

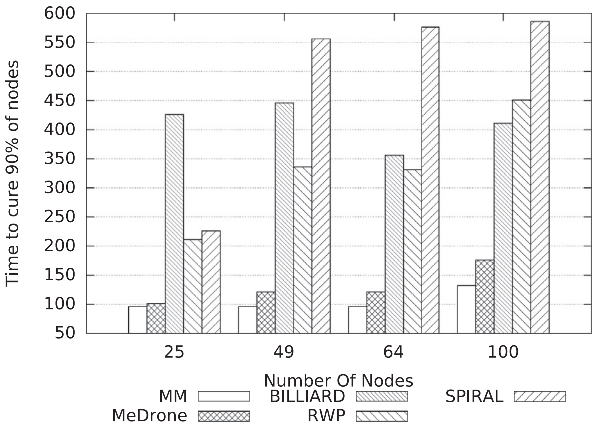

Fig. 5. Time to cure 90% of nodes. *Gaussian* Topology.

a heuristic that provides a sub-optimal movement scheme using only the information available locally at each node in the network. The proposed idea is to move the *searcher* continuously towards the regions with the highest density of infected nodes.

In this paper we assume that all the nodes have an equal and well known communication range R and that each node is capable of identifying and counting the infected nodes within it, thanks to a series of distress messages sent out by the IDS of the compromised neighbors [26]. This number will be referred in the following as the *number of Known Infected Nodes* nKIN and it is enough to estimate the density of infected nodes $D_{infected}$ in a circular area of which the node is the center by the following formula $D_{infected} = \frac{nKIN}{\pi \cdot R^2}$. Two concurrent processes compose the Distributed Curing Algorithm (DCA) proposed in this paper: (i) an *Alerting Process* during which the information about the malware diffusion are published by the infected nodes and propagated through the network and a (ii) *Healing Process* where the nodes are actually cured. The last process is carried out by the *searcher* and is composed of two continuously repeated phases. The first one is the *Positioning* phase, in which the *searcher*, aware of the epidemic from the *Alerting Process*, plans its movements. The second one is the *Curing* phase, where the *searcher* accesses the infected nodes and restores them. The processes can be handled by internal timers into the nodes without needing any synchronization.

5.1. The alerting process

As mentioned in Section 3, all messages for the cure are carried in Layer-2 frames and broadcasted by nodes through their short-range radio interface.

Whenever the IDS discovers the presence of a compromised component of its node, it begins the *Alerting Process* and sends an *SOS* message every Δt_{SOS} seconds on the node's short range interface.

Any *SOS* message contains:

1. **Sender hash (*SiD*)**: A variable-length label that identifies both the node sending the message locally and the original issuer.
2. **Sender position (*SP*):** 64 bit long field containing the absolute position (GPS) of the node generating the *SOS*. Given that in our scenario the nodes are stationary this information can be stored in the secure memory at time of node installation. According to [40], the absolute position can be substituted by a relative one, using the *SOS* packets to build the necessary infrastructure.
3. **The number of known infected nodes (*nKIN*):** Representing the perceived density of infected neighbors. The number of *SOS* messages heard in a reference time interval $\Delta t_{observ} = 2\Delta t_{SOS}$ can be an indication of this density. If no other *SOS* message has been received, the field is set to one. The timings were set according to preliminary simulations and out previous works on the topic [41].

Every node that receives the *SOS*, rebroadcasts it once for each reception from the original sender, using the hashes to discern duplicates within a listening Δt_{observ} time frame. If, during this interval, more than one message is received, only the one with the highest *nKIN* is stored for forwarding. The local *nKIN* is considered also in the ranking.

5.2. The healing process

As already mentioned, the Healing Process is composed of two phases, namely the (i) *Positioning* and the (ii) *Curing* ones. In the *Positioning* phase, we suppose that the *searcher* roams the area where the nodes are deployed, constantly monitoring its network interfaces, listening to any incoming *SOS* packet. Whenever one of these messages is received, the *searcher* extracts the *SP* information and ranks the senders based on the nKIN. extracts and ranks the *SP* location. Previous work results [41] show that any information, propagating from node i to node j, has to travel twice the number of hops between i and j. According to this result, we have set the *movement interval* T_m equal to the network diameter, in terms of number of hops, mulitplied by Δt_{SOS}. After the *movement interval* expires, the *searcher* selects the *SP* with the highest *nKIN* and starts the movement towards it. In the following we name this movement scheme as *MeDrone* (which stands for Medical Drone). Once the *searcher* arrives to the selected SP, it begins the curing phase (phase 2). To start the *Curing* phase, the *searcher* advertises its presence to all the nodes in its communication range by transmitting *Hello* messages every Δt_{Hello} seconds. The *Hello* message consists of a set of identifiers for the IDS of the nodes to recognize the *searcher*. The *cure* diffusion scheme has been devised as an exchange of data between the IDS of the infected nodes and the searcher. Data is directly carried in layer 2 frames. Once the data exchange is over, each formerly infected node has been provided with a clean image of their operating system. This image is then used by the IDS to overwrite the compromised parts of the operating system. The image is transferred in signed chunks. The process is driven by the exchange of *Interest* and *Data* packets between the infected node and the *searcher*, respectively. The *Interest* is broadcasted by the infected node upon receiving a Hello message to request a specific set of chunks. Nominally the *searcher* waits for a variable interval (the *Interest Interval*, set according to the number of sensed neighbors) to collect all the *Interests*, sort the requests and then send only one copy of the information requested in a burst of chunks during the variable *Chunk Interval*. In case of a packet loss, the *Interest* is retransmitted with its values properly adjusted to request the missing pieces, until all pieces are received. To reduce the collision probability from multiple simultaneous transmissions (that would happen, for instance, when multiple nodes respond to the same *Interest* request) each node i waits a random defer time T_D before transmitting each burst of chunks through *Data* packets. During this interval, the node i checks the channel to detect other nodes' transmission that are responding to any other requests. If no other node is transmitting, then i sends the burst. When the download is complete (no *Interest* packets are received for Δt_{Hello} seconds) the *searcher* returns in phase 1 and starts again to wait for *SOS* messages. At the end of the secured transfer the infected node is rebooted as healed. The transfer is called secure because each message is digitally signed to be recognized as valid by the IDS on the infected nodes [35]. We have chosen to fragment the image of the operating system diffused by the *searcher* in variable-length *chunks* at the Application Layer. This choice permits to change the chunk dimensions on-the-fly according to the L2 technology used.

6. Performance evaluation

In this Section, we compare the movement scheme computed by our algorithm, *MeDrone*, to four other movement schemes. In all five schemes the curing phase, described in Section 5, is the same, whereas they differ in the path chosen for the *searcher* to move through the WSN.

The first scheme of movement is to let the *searcher* move according to the Random Way Point model (*RWP*) [42]. This approach represents a simple yet effective scheme, where no effort is made to optimize the movements. The second and third alternatives consist in moving the *searcher* according to two predefined patterns, often used in the trajectory setting of planes and helicopters in order to cover the searching area during rescue operations: the *Spiral* and *Billiard* schemes [43]. In the *Spiral* scheme, the *searcher*, starting from any position in the network lattice, will move to the lattice's center and then start a rectangular spiral-like counter-clockwise movement pattern, where the separation distance between the spiral arms is constant and equal to twice the estimated node transmission range. In the *Billiard* scheme, the *searcher* node will initially position itself in a corner of the lattice and then move and rebound against the lattice's external boundaries. The angles of incidence with the boundaries are set to cover the whole area. Using this scheme, the *searcher* is present at the network lattice's boundaries longer than its center. The fourth alternative is to use the set of waypoints that come from the model of Section 4 and its implementation in Appendix A, which is used as an upper bound for the performance evaluation. In the following we will refer to it as the *MM* (Mathematical Model) scheme. It is worth to be noticed that the RWP, Billiard and Spiral schemes are general searching patterns that follows an *a-priori* scheme. They just require the searcher to know a rough estimate of the network lattice extension, without any hints about the position of the searched object. MeDrone, instead, thanks to the *alerting process*, has some knowledge, yet partial, of the position of the nodes to be healed. On the other hand, MM requires the complete a-priori knowledge of the network and nodes' characteristics, as well as the complete description of the diffusion dynamics. Our primary intent in selecting the competitors schemes was to demonstrate that, whichever the movement scheme, the flying drones could be effectively used to remove epidemics from a WSN. Then, in comparing the performance of MeDrone with RWP, Spiral and Billiard, we intended to evaluate the contribution of the *alerting process* and, specifically, how much improvement it offers against the position-agnostic movement schemes. Finally, in comparing MeDrone with MM, we intended to evaluate how far it is from the performance achievable by having the perfect knowledge about the nodes and the epidemics.

Table 2
802.11g simulation parameters.

PHY parameter	Value	MAC parameter	Value
Frequency	2.4 GHz	SlotTime	9 μs
Receiver Sensitivity	−86 dBm	SIFS	16 μs
Transmission Power	18 dBm	Preamble Length	96 bit
Max distance	32 m		

We have implemented *RWP*, *Spiral*, *Billiard* and *MeDrone* in the ns-2 [44] simulator, while the *MM* has been coded in the FICO Xpress Suite and Mosel language [45].

The reference simulated topologies are multiple: (i) a 300 × 300 m area where the nodes are placed according to a bi-dimensional Gaussian distribution originating from a corner (identified as *Gaussian*), with a standard deviation of 150m; (ii) a 160 × 160 m square area where a variable number of nodes are placed according to a regular grid pattern (identified as *Grid*) and (iii) a variable dimension area where the nodes are placed according to the realistic topology (identified as *Realistic*) generated using a specific framework called NPART [46]. NPART is capable of designing network topologies whose statistical characteristics are similar to the ones measured in real networks. The number of nodes in all the topologies varies as follows [25, 49, 64, 100]. Hence, the distances between each couple of nodes in the *Grid* vary accordingly [32, 23, 20, 16] m. The *Gaussian* topology is characterized by the presence of an agglomeration point for the nodes, i.e., the center of the distribution, which has a higher relative density than the rest of the area. In the *Grid* topology, instead, the nodes are evenly distributed. As specified in the literature [46], the *Realistic* topology is composed of a set of connected clusters of nodes around the center of the network area, with multiple areas of node concentration. These features influence how effectively the proposed algorithm can find the most dense areas of infected nodes. For instance, the *Grid* topology does not have areas with an increased relative density of nodes, being these infected or not, and the *Gaussian* topology has a single one.

All the nodes are equipped with a single IEEE 802.11 interface and a Ricean fading model takes into account the multi-path effects due to various obstacles. As shown in Table 2, Physical and MAC parameters are based on IEEE 802.11g. Transmission power and receiver sensitivity figures are taken from data-sheets of devices available on the market [47].

The *searcher* and the first set of infected nodes are initially placed at random positions within the network lattice using a

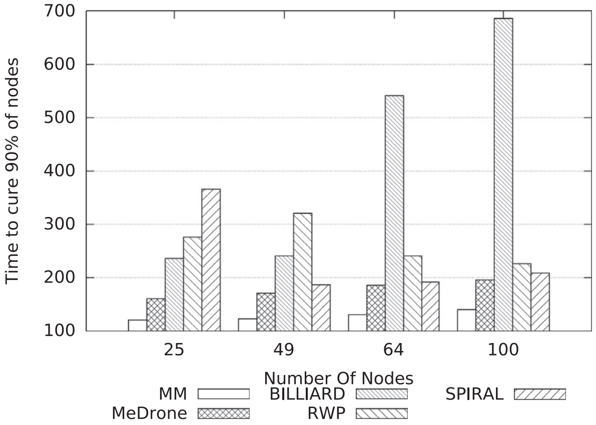

Fig. 6. Time to cure 90% of nodes *Grid* Topology.

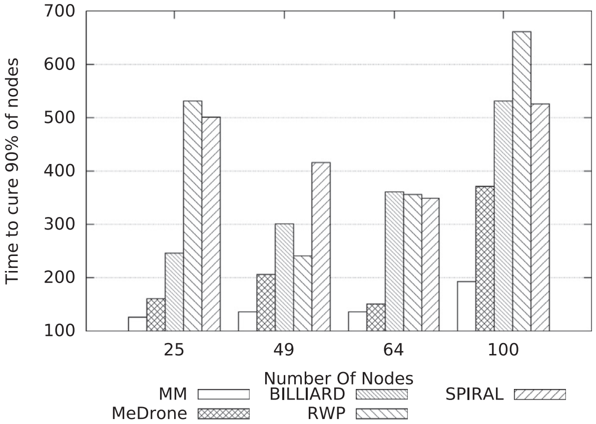

Fig. 7. Time to cure 90% of nodes *Realistic* Topology.

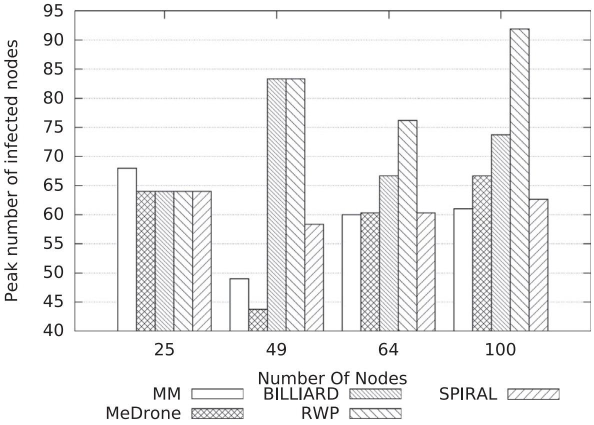

Fig. 8. Peak value of infected nodes. *Gaussian* Topology.

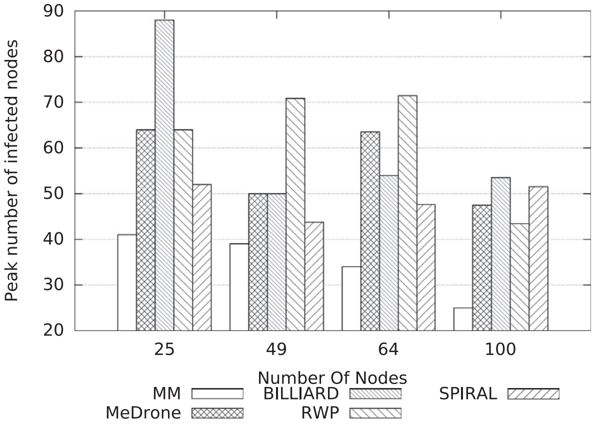

Fig. 9. Peak value of infected nodes. *Grid* Topology.

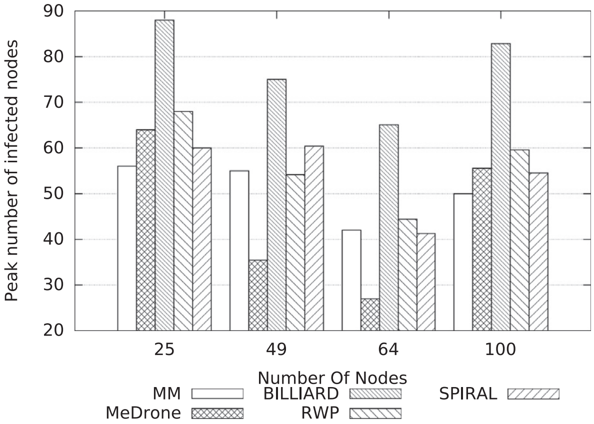

Fig. 10. Peak value of infected nodes. *Realistic* Topology.

uniform distribution. Once the simulation starts, the *searcher* moves according to the mobility pattern computed by the *RWP*, *MeDrone*, *Spiral*, *Billiard* or *MM* schemes. The *searcher* speed is set according to the literature in this field [48]. The results of simulation campaigns are presented in the following subsection.

Our main objective is to evaluate the evolution and the steady state number of infected nodes to understand which scheme gives the best results, in a given topology, for the following two metrics: (i) the capability of a scheme to limit the overall infection, measured as the peak number of infected nodes (*Peak Value*) and (ii) the time needed to cure the network, i.e., the time needed to reduce the number of infected nodes to the 10% of the total network nodes after the infection peak has been reached (Time To Settle, *TTS*). To give a clearer insight on how these values vary with topologies and movement schemes, they are reported separately in the last subsection, aggregating all the previous results in a wider view.

6.1. Outcomes of the simulation campaigns

Several simulation campaigns, varying the number of nodes in the network, have been performed. The first set of results considers a network of only 25 nodes. This first simulation campaign is meant to provide an insight on how each movement pattern works, when scarcely connected networks are considered. Successive simulation campaigns, which consider an increasing number of nodes in the network, have been conducted to investigate the scalability of the proposal as well as the impact of nodes density on the performance of the movement schemes. Results for higher number of nodes are summarized in figures from 5 to 7 and from 8 to 10, which show the average of time to settle (*TTS*) and of the *Peak Value* respectively.

As a general trend we may observe that, when the number of nodes in the network increases, the epidemic spreads quicker and reaches a greater *Peak Value*. This is due to the greater number of connection between adjacent nodes that the epidemic can use to propagate. Also, the *TTS* increases with the number of nodes. This is essentially due to: (i) the increased number of times the *searcher* must iterate the healing process (see Section 5.2) because of the increased number of infected nodes, (ii) the fact that the epidemic continues to spread while the *searcher* is applying the cure, and (iii) the fact that the increasing number of requests issued by nodes that struggle to be healed (see Section 5.1) tend to congestionate the communication channel thus slowing the healing process.

Following our evaluation metrics, the presence of an area with a large concentration of nodes in the *Gaussian* topology (Fig. 2) gives the *MM* and *MeDrone* schemes a conspicuous advantage over the others in terms of *TTS*, as they are capable of exploiting the density increase in a limited area. Driving the *searcher* across empty areas at the lattice boundaries the *Billiard* gives instead poor results. In this topology the *Spiral* scheme performances suffers highly in the 100-nodes scenario while for lowest density it is capable to outperform the *Billiard* (Fig. 5).

In the *Grid* topology the *MM* scheme retains again the lowest *Peak Value* and the lowest *TTS*, while *MeDrone* still obtain the second lowest *TTS* apart from the one obtained by *MM*, thanks to the capability to track down infections (Fig. 3). Nonetheless, this is the scenario where the proposed scheme shows fewer improvements compared to its competitors. This is due to the fact that *MeDrone* cures the infection by starting from the most dense region in the network while a *Grid* topology does not present such a region. Hence, the performance of the other movements schemes, particularly those of the *Spiral* and *RWP* schemes, are close to that of the proposed scheme and sometimes present a lower *Peak Value*. It could be further noticed, that the *Billiard* scheme behaves particularly bad in this scenario. Specifically, as the number of nodes increases (Fig. 6), the performance of the *Billiard* scheme quickly worsens at such an extent that, for the highest number of nodes, it fails to completely eradicate the infection. This behavior is due to the intrinsic movement features of the scheme. In fact, in this case the *searcher*, traveling back and forth between the network area boundaries, occupies the borders of the lattice for longer than the lattice's center. In this way, it has less time to contact the nodes in the middle. Thus, in that area the infection is less constrained. In this topology, (Fig. 6) the *Spiral* output is almost independent from the density whereas in the *Realistic* one (Fig. 7) the *TTS* decreases with the number of nodes to increase again for the 100 nodes case. The *RWP* scheme instead shows an improvement of the *TTS* as density increases and a degradation in the *Gaussian* one as the same value decreases. In the *Realistic* topology (Fig. 4), in respect to *Spiral*, *Billiard* and *RWP* schemes, the *MeDrone* performs the best in terms both of *Peak Value* and of *TTS*. The *searcher* can move among the set of connected *clusters* of this topology [46], and briefly distribute the cure.

Figs. 8–10 summarize the trends illustrated so far. Analyzing the *Peak Value*, we show that, at least for the *Gaussian* and *Realistic* topologies, the *MeDrone* scheme is the best option as it is capable of reaching the various areas dense of infected nodes and quickly cure them. The only exception to this behavior, also shown in the previous set of figures, is for high total number of nodes, where the tracking algorithm is slow in identifying the areas with the highest number of infected nodes. In the *Grid* topology (Fig. 9), the epidemic diffuses among evenly-placed nodes, thus the *MeDrone* cannot easily detect the target areas for low densities and, when density is higher, the sheer number of nodes makes the very general epidemic diffusion difficult. In the same topology the *Billiard* and *Spiral* schemes fare reasonably the same. Results are different for the *Gaussian* topology (Fig. 8), where the presence of a unique large concentration point close to the center supports better the *MeDrone* and *Spiral* schemes. For the *Realistic* topology (Fig. 10) the best choice, apart from the *MM*, is again the *MeDrone*. The peculiar features of the node placement scheme, and the presence of various clusters, favors the *RWP* scheme instead of the *Billiard* one. In this last topology, the greatest concentration of nodes is near the center of the lattice, therefore the *Spiral* scheme outperforms the *Billiard* and the *RWP* schemes that waste time traveling through empty areas. The *Billiard* is again the worst performer. As a result, none of the *Billiard*, *RWP* and *Spiral* movement schemes can be considered as a valid alternative to *MeDrone* because they are all unable to guarantee a constant and good performance over a wide range of topologies and nodes densities.

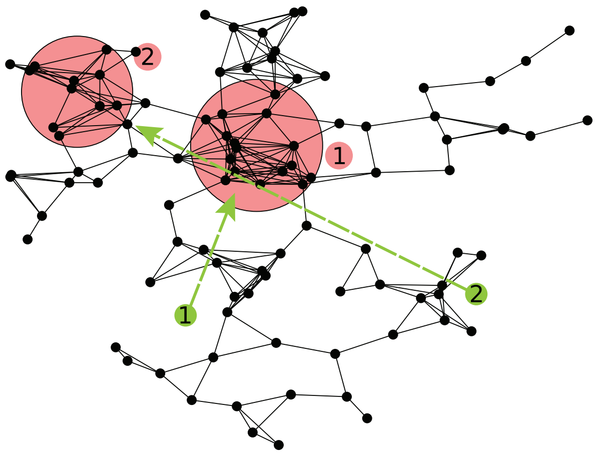

Fig. 11. Multiple node movement example.

7. An introduction on multiple searcher coordination

Up to this point, in this paper, we have considered just one *searcher* node. Increasing the number of mobile elements and introducing coordination is an important step to create a more efficient and complete framework. Unfortunately, the model developed so far for the TCP only includes the possibility of finding the optimal solution for a single *searcher* and an actual coordination environment between entities devoted to cure a network is beyond the scope of this paper. However, it is possible to initially evaluate how the network and the epidemic spread will react in presence of a very simple coordination scheme. For achieving this feature, we modified our previous algorithm for the identification of the most dense areas of infected nodes in order to keep track of a set of points instead of a single one. Specifically, the framework can keep track of the two most dense geographical areas and diffuse the information about them. We further considered two *searchers*. At each time the first *searcher* is committed to cure the most dense area while the second *searcher* will take care of the second most dense area (see Fig. 11). If a single aggregation of infected node is detected, then both the *searchers* converge on that region. Following this approach we got a first promising outlook about the feasibility and the convenience to extend *MeDrone* to a scenario where a group of drones is deployed to cure an infected network. To evaluate the results we used the same simulation environment of the Section 6 and we focused on the scenario with 100 nodes, in order to have a dispersed network where a single *searcher* has demonstrated some difficulties in following the epidemic. Figures from 12,13,14,15,16,17,18,19 to 20 show the evolution of the epidemic comparing the single and the multi-node solutions for respectively the *Gaussian*, the *Grid* and the *Realistic* topologies. In detail, for each figure it is displayed the mutation dynamics of the *MeDrone*, together with each one of the other schemes for both one and two *searchers*. For the 2-*searcher* version the other schemes are modified as follows: for the *RWP*, a second set of random way-points are followed by the second *searcher*; for the *Billiard*, the starting point for the second *searcher* is set on the other side of the network area and the rebound points are mirrored in respect to the original trajectory; finally, for the *Spiral*, the second *searcher* travels clockwise starting from the same point of the first one. For all the topologies, the *MeDrone* scheme shows an improvement in both

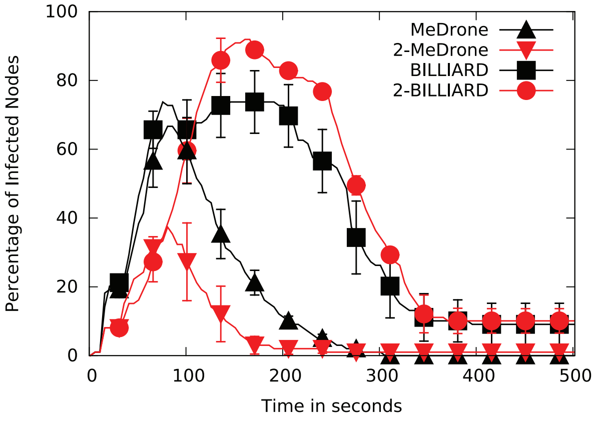

Fig. 12. Percentage of infected elements for the *Gaussian* topology. *Billiard* Scheme.

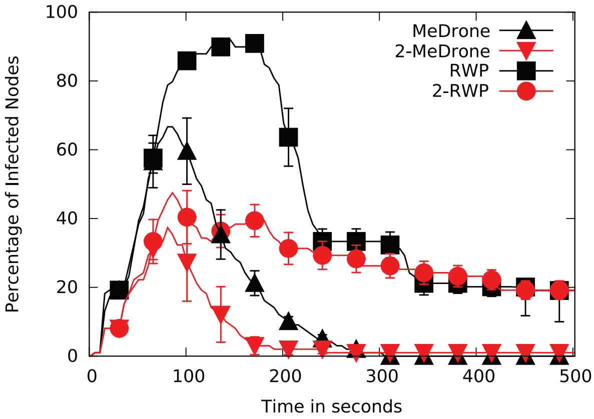

Fig. 13. Percentage of infected elements for the *Gaussian* topology. *RWP* Scheme.

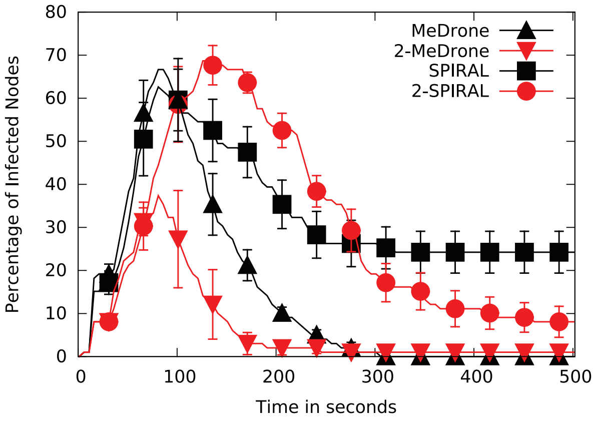

Fig. 14. Percentage of infected elements for the *Gaussian* topology. *Spiral* Scheme.

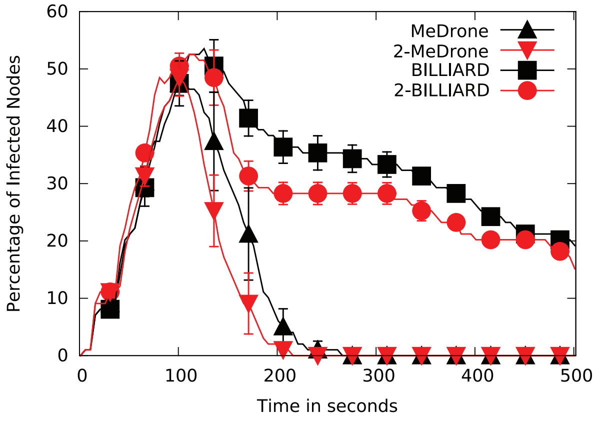

Fig. 15. Percentage of infected elements for the *Grid* topology. *Billiard* Scheme.

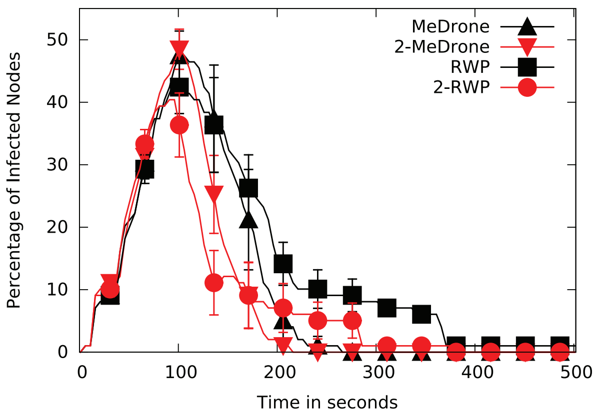

Fig. 16. Percentage of infected elements for the *Grid* topology. *RWP* Scheme.

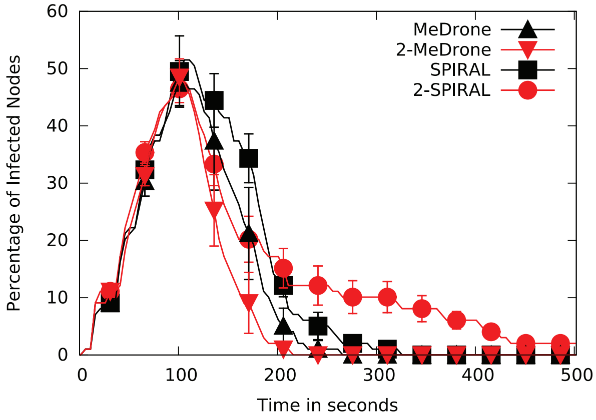

Fig. 17. Percentage of infected elements for the *Grid* topology. *Spiral* Scheme.

in the time necessary to eradicate the infection (*TTS*) and the *Peak Value*. Important results can be obtained analyzing the behaviors of different schemes in the different topologies. In Figures from 12 to 14 the results for the *Gaussian* topology are shown. The presence of multiple *searchers* driven by the *MeDrone* scheme decreases the spread of the malware better than the other schemes. This happens because the two *searchers* are both driven to the center of the Gaussian where the infected node density is the highest.

This maximizes the initially covered area and quickly clean the infection. The improvements introduced by the 2-*searcher* version of the other schemes in the same topology are not as significant. In Fig. 12 and 14 it is shown that, for the *Billiard* and *Spiral* schemes, when the two *searchers* move at the same time in the very same area, the values of *TTS* and *Peak Value* do actually increase.

Instead, if the movement of the two *searchers* is loosely decoupled as in the *RWP* (Fig. 13), there is an improvement.

As for the case of the single *searcher*, the *Grid* topology is the most challenging one for the *MeDrone* scheme (see figs. from 15 to 17). Even if it is possible to highlight an improvement when two *searchers* are deployed, the actual gains are limited.

This is mainly due to the dispersion of the nodes and the time spent by the *searchers* waiting for an area to be cured before moving to the next one. As shown in the previous sections, the *Billiard* scheme is extremely dispersive for the *Grid* topology (Fig. 15). In

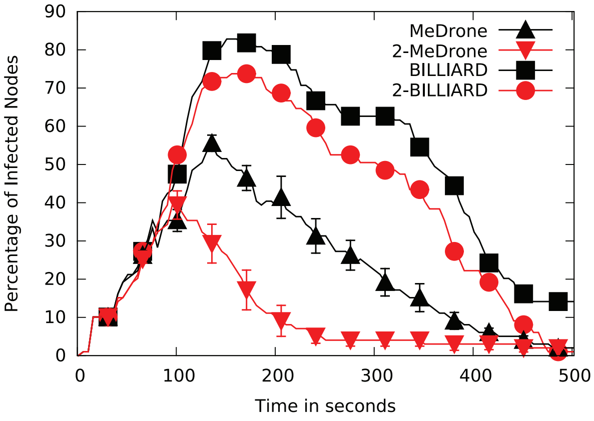

Fig. 18. Percentage of infected elements for the *Realistic* topology. *Billiard* Scheme.

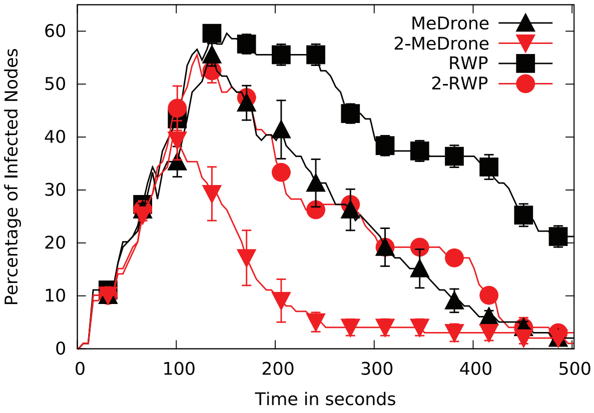

Fig. 19. Percentage of infected elements for the *Realistic* topology. *RWP* Scheme.

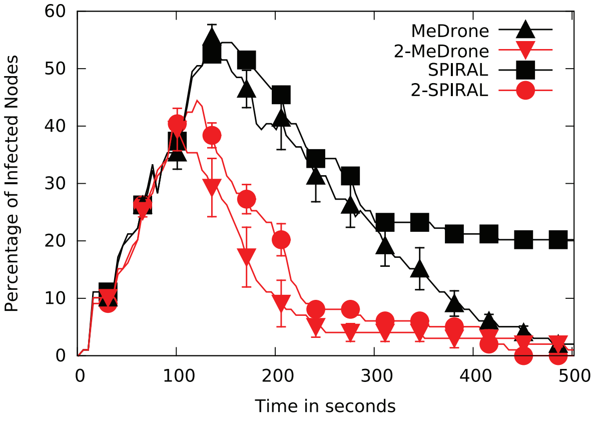

Fig. 20. Percentage of infected elements for the *Realistic* topology. *Spiral* Scheme.

this case the *RWP* and *Spiral* schemes show similar results as the *MeDrone* with 1 and 2 *searchers* (Fig. 16).

The results relevant to the *Realistic* are shown in Figures from [18,19] to 20. The presence of multiple aggregation points with different densities (i.e. different clusters of nodes) allows the 2-node *MeDrone* scheme to achieve the best improvements among all the topologies. Selecting different areas and serving them in parallel shows that the *Peak Value* and *TTS* are extremely reduced (roughly one third of nodes becomes infected and half the time is necessary to cure 90% of nodes) using the simple scheme just described. The *MeDrone* scheme outclasses the *Billiard* with both 1 and 2 *searchers* (Fig. 18); the *RWP* (Fig. 16) with two *searchers* and it is slightly better than the *Spiral* in the same situation (Fig. 17). We underline again that the *Spiral*, *Billiard*and *RWP* schemes need the information about the network area extension to calculate their starting points. In the same situation, the *MeDrone* is capable to correctly dispatch the searchers by only using the information carried by the *SOS* messages. These results open the way to research on a practical implementation of a coordination system between nodes that can bring significant benefits to the management of proximity infections in WSNs.

8. Conclusions and future works

In this paper we addressed the issue of facing a malware spreading in a WSN. Using the concepts of controlled mobility and information diffusion we defined an algorithm (i) to notify the spreading of the malware, (ii) to lead an autonomous flying robot, which can cure infected nodes, along a path through the WSN and (iii) to heal the infected nodes. To benchmark the proposed algorithm, we developed a mathematical model that defines the best path to be followed by the flying robot to cure the infected nodes as quick as possible. Using the results obtained from the mathematical model as the upper bound and the ones from the application of random way-point movement as the lower bound, we have assessed the satisfactory performance of the proposed algorithm. Finally, we have given an outlook on the directions to extend the proposed algorithm by using multiple *searchers*. As a future work we intend to further address the use of multiple *searchers* and to develop a larger and more comprehensive mathematical model, in order to provide more generalized solutions, which can be applied in real scenarios. We expect also to address the creation of a coordination framework based on solutions that do not rely on absolute coordinate systems.

Acknowledgement

This work has been carried out in the framework of the DIVINA Challenge Team, which is funded under the Labex MS2T program. Labex MS2T is supported by the French Government, through the program “Investments for the future” managed by the National Agency for Research (Reference ANR-11-IDEX-0004-02).

Appendix A. Mathematical formulation

In the following subsections we describe a linear programming formulation for the problem that only involves binary variables.

Parameters

All the parameters of the model are provided in Table A.3.

Regarding the last parameter, we remark that a node, *is in possession of the cure* if the *searcher* is in the same place as the node.

The content of the set of wireless links and other parameters come from a preliminary simulation campaign in which were analyzed the behaviors of the nodes. The aforementioned set was filled from the adjacency matrix of the various topologies considered in Section 6 while the times necessary for the *searcher* to physically travel are computed according to the average speeds of flying robots.

As the modification of the infection dynamics occurs as the *searcher* moves, the parameter T_m (movement interval) is the smallest time unit considered for the model and we define the others using the former as a measurement unit. In this way the parameter T_t, comes from an analysis of the time needed to complete the transfer in a preliminary set of simulations, quantized in T_m units. The parameter T_a, that describes the time needed for the *searcher* to access the secured part of an infected node S.O. is set again accordingly but, as in the simulations it was almost negligible, we set it as the same as T_m.

Table A.3
Data.

T_m	Movement interval, the running time needed to choose a new direction for the *searcher*
T_t	Running time for a node to download a block of software representing either the cure or the malware
T_a	Running time for the *searcher* to access the secured part of a node operating system
N	Set of all nodes
S	Set of possible states for each node
E	Set of wireless links, bidirectional
E_c	Set of physical paths, bidirectional
d_e	Time needed to travel along the edge e, in time units
d_i	Total time needed for a node i in time units to complete the transfer of a module
d_{ij}	Time needed for a *searcher* to travel from node i to j
N^*	Subset of nodes initially infected
t^*	Time unit at which the *searcher* enters the network the network
i^*	Node at which the *searcher* is inserted

Table A.4
State and decision variables.

x_{si}^t	Equal to 1 iff node i is in state $s \in \{1, 3, 5\}$ at the beginning of period t
x_{si}^{pt}	Equal to 1 iff node i is in state $s \in \{2, 4\}$, since p time units, at the beginning of period t
y_i^t	Equal to 1 iff node i carries the *searcher* at the beginning of period t
z_{ij}^t	Equal to 1 iff the *searcher* is starting to travel from i to j at the beginning of period t

Table A.5
Conditional variables.

Y_i^t	Equal to 1 iff node i is connected to the *searcher*
M_i^t	Equal to 1 iff node i is connected to at least a node in state 3 and not to the *searcher*
C_i^t	Equal to 1 iff node i is connected to at least a node in state 1, 2 or to the *searcher*
S_i^t	Equal to 1 iff node i is not connected to nodes in states 3 and not connected to the *searcher*
X^t	Equal to 1 iff at least one node is infecting or malicious at the beginning of period t

Variables

We classify the optimization variables in two groups. The first group (Table A.4) contains the variables of the problem, describing, for each time unit, the state of each node and the *searcher* movements. The second group (Table A.5) describes the variables imposing conditions in the model.

Objective function and constraints

The purpose of the objective function is to minimize the incurrence of the time unit at which there are no more malicious or infected nodes left.

$$\min \quad \sum_{t\in T} X^t.$$

The constraints of the model are separated into two groups:

1. Constraints of Eq. (A.1), ensure that the problem variables take the admissible values.
2. Constraints of Eq. (A.2) are a mathematical translation of the transitions of Fig. 1.

Namely, the constraints translate the aforementioned logical conditions into linear constraints. Taking for instance conditional variable Y_i^t, the first two constraints imply that Y_i^t takes value one if the *searcher* is located at node i in period t or connected to i through an edge in E. Then, the third constraint implies that the variable takes value zero if the *searcher* is neither located at node i nor connected to it through an edge.

$$\begin{aligned}
Y_i^t &\geq y_i^t && i \in N, t \in T\\
Y_i^t &\geq y_j^t && ij \in E, t \in T\\
Y_i^t &\leq y_i^t + \sum_{ij\in E} y_j^t i \in N, t \in T\\
M_i^t &\geq x_{3j}^t - y_i^t - \sum_{ij'\in E} y_{j'}^t ij \in E, t \in T\\
M_i^t &\leq \sum_{ij\in E} x_{3j}^t t \in T\\
M_i^t &\leq 1 - y_i^t - \sum_{ij\in E} y_j^t t \in T\\
C_i^t &\geq x_{1j}^t && ij \in E, t \in T\\
C_i^t &\geq x_{2j}^{pt} \; ij \in E, t \in T, p = 0, \ldots, d_j - 1\\
C_i^t &\geq x_{5j}^t && ij \in E, t \in T\\
C_i^t &\geq y_i^t && i \in N, t \in T\\
C_i^t &\geq y_j^t && ij \in E, t \in T\\
C_i^t &\leq y_i^t + \sum_{ij\in E}\left(y_j^t + x_{1j}^t + x_{5j}^t + \sum_{p=0}^{d_i-1} x_{2j}^{pt}\right)\\
&\qquad t \in T\\
S_i^t &\leq 1 - y_i^t && i \in N, t \in T\\
S_i^t &\leq 1 - y_j^t && ij \in E, t \in T\\
S_i^t &\leq 1 - x_{3j}^t && ij \in E, t \in T\\
S_i^t &\geq 1 - \sum_{ij\in E}(y_j^t + x_{3j}^t) - y_i^t t \in T\\
X^t &\geq \sum_{i\in N} \frac{x_{2i}^t + x_{3i}^t}{2|N|} t \in T
\end{aligned} \tag{A.1}$$

$$\begin{aligned}
x_{1i}^t &= x_{1i}^{t-1} S_i^{t-1} && i \in N, t \in T^*\\
x_{1i}^t &= x_{1i}^{t-1} && i \in N, t \in T\setminus T^*\\
x_{2i}^{0t} &= x_{2i}^{0t-1} S_i^{t-1} && i \in N, t \in T\setminus T^*\\
x_{2i}^{0t} &= x_{2i}^{0t-1} S_i^{t-1} + x_{1i}^{t-1} M_i^{t-1} && i \in N, t \in T^*\\
x_{2i}^{pt} &= x_{2i}^{pt-1} S_i^{t-1} + x_{2i}^{p-1t-1} M_i^{t-1}\\
&\qquad p = 1, \ldots, d_i - 1, i \in N, t \in T\\
x_{3i}^t &= x_{3i}^{t-1} + x_{2i}^{d_i-1t-1} M_i^{t-1} && i \in N, t \in T\setminus T^*\\
x_{3i}^t &= x_{3i}^{t-1}(1 - Y_i^{t-1}) + x_{2i}^{d_i-1t-1} M_i^{t-1} \; i \in N, t \in T^*\\
x_{4i}^{0t} &= x_{4i}^{0t-1}(1 - Y_i^{t-1}) && i \in N, t \in T\setminus T^*\\
x_{4i}^{0t} &= x_{4i}^{0t-1}(1 - Y_i^{t-1}) + x_{3i}^{t-1} Y_i^{t-1} && i \in N, t \in T^*\\
x_{4i}^{pt} &= x_{4i}^{pt-1}(1 - C_i^{t-1}) + x_{4i}^{p-1t-1} C_i^{t-1}\\
&\qquad p = 1, \ldots, d_i - 1, i \in N, t \in T\\
x_{5i}^t &= x_{5i}^{t-1} + x_{4i}^{d_i-1t-1} C_i^{t-1} && i \in N, t \in T\setminus T^*\\
x_{5i}^t &= x_{5i}^{t-1} + \left(x_{1i}^{t-1} + \sum_{i=0}^{d_i-1} x_{2i}^{pt-1}\right) Y_i^{t-1} + x_{4i}^{d_i-1t-1} C_i^{t-1}\\
&\qquad i \in N, t \in T^*
\end{aligned} \tag{A.2}$$

The equations of (A.2) are non-linear.

However, as the involved variables are all binary, we can use a well-known technique to linearize their products. Each product between any binary variable $v_1 v_2$ in the above equation can yield a new *real* variable *res* that satisfies Eq. A.3.

$$\begin{aligned} res &\le v_1 \\ res &\le v_2 \\ res &\ge v_1 + v_2 - 1. \end{aligned} \tag{A.3}$$

For the sake of simplicity, we do not rewrite explicitely the linearized constraints.

The *searcher*'s movement is formalized in Eq. A.4.

$$\begin{aligned} y_i^t &= 0 \qquad i \in N, t \le t^* - 1 \\ y_{i^*}^{t^*} &= 1 \\ y_i^{t^*} &= 0 \qquad i \in N \backslash i^* \\ y_i^t &= \sum_{ij \in E_c} z_{ji}^{t-d_{ij}} + y_i^{t-1} - \sum_{ij \in E_c} z_{ij}^{t-1} \\ & \qquad t \in T \\ \sum_{ij \in E_c} z_{ij}^t &\le y_i^t \qquad ij \in E_c, t \in T \end{aligned} \tag{A.4}$$

Ending the discussion, the constraints of Eq. A.5 describe the system's starting conditions, and enforce that all optimization variables be {0, 1}-valued.

$$\begin{aligned} x_{1i}^0 &= 1 \qquad i \in N \backslash N^* \\ x_{3i}^0 &= 1 \qquad i \in N^* \\ x_{1i}^0 + x_{3i}^0 + x_{5i}^0 + \sum_{p=0}^{d_i-1} (x_{2i}^{p0} + x_{4i}^{p0}) &= 1 \\ & \qquad i \in N \\ Y, M, C, S, X, x, y, z &\in \{0, 1\} \end{aligned} \tag{A.5}$$

References

[1] M. La Polla, F. Martinelli, D. Sgandurra, A survey on security for mobile devices, IEEE Commun. Surv. Tut. 15 (1) (2013) 446–471.

[2] Y. Wang, S. Wen, Y. Xiang, W. Zhou, Modeling the propagation of worms in networks: A survey, IEEE Commun. Surv. Tut. 16 (2) (2014) 942–960.

[3] B. Sun, G. Yan, Y. Xiao, Worm propagation dynamics in wireless sensor networks, in: Communications, 2008. ICC'08. IEEE International Conference on, IEEE, 2008, pp. 1541–1545.

[4] I.F. Akyildiz, T. Melodia, K.R. Chowdury, Wireless multimedia sensor networks: a survey, Wireless Commun. IEEE 14 (6) (2007) 32–39.

[5] J.W. Lartigue, C. McKinney, R. Phelps, R. Rhodes, A.D. Rice, A. Ryder, A tablet-controlled, mesh-network security system: an architecture for a secure, mesh network of security and automation systems using arduino and zigbee controllers and an android tablet application, in: Proceedings of the 2014 ACM Southeast Regional Conference, in: ACM SE '14, ACM, New York, NY, USA, 2014, pp. 33:1–33:4, doi:10.1145/2638404.2638500.

[6] W. Shengjun, C. Junhua, Modeling the spread of worm epidemics in wireless sensor networks, in: Wireless Communications, Networking and Mobile Computing, 2009. WiCom'09. 5th International Conference on, IEEE, 2009, pp. 1–4.

[7] T.M.I., Symbos cabir.a., 2004. URL http://about-threats.trendmicro.com/us/archive/malware/SYMBOSCABIR.A.

[8] T.M.I., Ios ikee.a., 2006. URL http://www.trendmicro.com/vinfo/us/threat-encyclopedia/malware/IOSIKEE.A.

[9] X. Wang, Z. He, X. Zhao, C. Lin, Y. Pan, Z. Cai, Reaction-diffusion modeling of malware propagation in mobile wireless sensor networks, Sci. China Inf. Sci. 56 (9) (2013) 1–18.

[10] B. Sun, G. Yan, Y. Xiao, T. Andrew Yang, Self-propagating mal-packets in wireless sensor networks: dynamics and defense implications, Ad Hoc Netw. 7 (8) (2009) 1489–1500.

[11] F. Li, Y. Yang, J. Wu, CPMC: an efficient proximity malware coping scheme in smartphone-based mobile networks, in: INFOCOM, 2010 Proceedings IEEE, IEEE, 2010, pp. 1–9.

[12] A. Houmansadr, S.A. Zonouz, R. Berthier, A cloud-based intrusion detection and response system for mobile phones, in: Dependable Systems and Networks Workshops (DSN-W), 2011 IEEE/IFIP 41st International Conference on, IEEE, 2011, pp. 31–32.

[13] Y. Nadji, J. Giffin, P. Traynor, Automated remote repair for mobile malware, in: Proceedings of the 27th Annual Computer Security Applications Conference, ACM, 2011, pp. 413–422.

[14] N.R. Zema, E. Natalizio, M. Poss, G. Ruggeri, A. Molinaro, Healing wireless sensor networks from malicious epidemic diffusion, in: Distributed Computing in Sensor Systems (DCOSS), 2014 IEEE International Conference on, IEEE, 2014, pp. 171–178.

[15] T.H. Chung, G.A. Hollinger, V. Isler, Search and pursuit-evasion in mobile robotics, Auton. Robots 31 (4) (2011) 299–316.

[16] M.A. Vieira, R. Govindan, G.S. Sukhatme, Scalable and practical pursuit-evasion with networked robots, Intell. Serv. Robot. 2 (4) (2009) 247–263.

[17] S. LaValle, D. Lin, L. Guibas, J.-C. Latombe, R. Motwani, Finding an unpredictable target in a workspace with obstacles, in: Robotics and Automation, 1997. Proceedings., 1997 IEEE International Conference on, 1, 1997, pp. 737–742 vol.1, doi:10.1109/ROBOT.1997.620123.

[18] M.E. Newman, Spread of epidemic disease on networks, Phys. Rev. E. 66 (1) (2002) 016128.

[19] E. Verriest, F. Delmotte, M. Egerstedt, Control of epidemics by vaccination, in: American Control Conference, 2005. Proceedings of the 2005, IEEE, 2005, pp. 985–990.

[20] R. Pastor-Satorras, A. Vespignani, Epidemics and immunization in scale-free networks, Handbook of graphs and networks: from the genome to the internet (2005) 111–130.

[21] G. Yan, H.D. Flores, L. Cuellar, N. Hengartner, S. Eidenbenz, V. Vu, Bluetooth worm propagation: mobility pattern matters!, in: Proceedings of the 2nd ACM symposium on Information, computer and communications security, in: ASIACCS '07, ACM, New York, NY, USA, 2007, pp. 32–44, doi:10.1145/1229285.1229294.

[22] S. Tang, A modified epidemic model for virus spread control in wireless sensor networks, in: Global Telecommunications Conference (GLOBECOM 2011), 2011 IEEE, IEEE, 2011, pp. 1–5.

[23] L. Sander, C. Warren, I. Sokolov, C. Simon, J. Koopman, Percolation on heterogeneous networks as a model for epidemics, Math. Biosci. 180 (2002) 293–305, doi:10.1016/S0025-5564(02)00117-7. URL http://www.sciencedirect.com/science/article/pii/S0025556402001177.

[24] V.M. EguíLuz, K. Klemm, Epidemic threshold in structured scale-free networks, Phys. Rev. Lett. 89 (10) (2002) 108701, doi:10.1103/PhysRevLett.89.108701.

[25] S. Brown, C.J. Sreenan, Software updating in wireless sensor networks: a survey and lacunae, J. Sensor Actuat. Netw. 2 (4) (2013) 717–760.

[26] A. Shabtai, U. Kanonov, Y. Elovici, Intrusion detection for mobile devices using the knowledge-based, temporal abstraction method, J. Syst. Softw. 83 (8) (2010) 1524–1537, doi:10.1016/j.jss.2010.03.046.

[27] P. Wang, M.C. González, C.A. Hidalgo, A.-L. Barabási, Understanding the spreading patterns of mobile phone viruses, Science 324 (5930) (2009) 1071–1076.

[28] M. Khouzani, S. Sarkar, E. Altman, Maximum damage malware attack in mobile wireless networks, IEEE/ACM Trans. Netw. (TON) 20 (5) (2012) 1347–1360.

[29] G. Yan, G. Chen, S. Eidenbenz, N. Li, Malware propagation in online social networks: nature, dynamics, and defense implications, in: Proceedings of the 6th ACM Symposium on Information, Computer and Communications Security, in: ASIACCS '11, ACM, New York, NY, USA, 2011, pp. 196–206, doi:10.1145/1966913.1966939.

[30] S.-M. Cheng, W.C. Ao, P.-Y. Chen, K.-C. Chen, On modeling malware propagation in generalized social networks, Commun. Lett. IEEE 15 (1) (2011) 25–27, doi:10.1109/LCOMM.2010.01.100830.

[31] W. Peng, F. Li, X. Zou, J. Wu, Behavioral malware detection in delay tolerant networks, IEEE Trans. Parallel Distrib. Syst. 25 (1) (2014) 53–63.

[32] J. Cheng, S.H. Wong, H. Yang, S. Lu, Smartsiren: virus detection and alert for smartphones, in: Proceedings of the 5th international conference on Mobile systems, applications and services, ACM, 2007, pp. 258–271.

[33] A.-F. Sui, D.-F. Guo, T. Guo, M.-z. Li, Network behavior based mobile virus detection, in: Communication Technology (ICCT), 2012 IEEE 14th International Conference on, IEEE, 2012, pp. 872–876.

[34] C. Gao, J. Liu, Modeling and restraining mobile virus propagation, IEEE Trans. Mobile Comput. 12 (3) (2013) 529–541.

[35] A.U. Schmidt, N. Kuntze, M. Kasper, On the deployment of mobile trusted modules, in: Wireless Communications and Networking Conference, 2008. WCNC 2008. IEEE, IEEE, 2008, pp. 3169–3174.

[36] J. Grossschadl, T. Vejda, D. Page, Reassessing the TCG specifications for trusted computing in mobile and embedded systems, in: Hardware-Oriented Security and Trust, 2008. HOST 2008. IEEE International Workshop on, IEEE, 2008, pp. 84–90.

[37] O. Aciicmez, A. Latifi, J.-P. Seifert, X. Zhang, A trusted mobile phone prototype, in: Consumer Communications and Networking Conference, 2008. CCNC 2008. 5th IEEE, IEEE, 2008, pp. 1208–1209.

[38] TCG, TPM Main Part 1 Design Principles, 2011. Specification Version 1.2 Revision 116.

[39] R.B. Allan, R. Laskar, On domination and independent domination numbers of a graph, Disc. Math. 23 (2) (1978) 73–76.

[40] N. Zema, N. Mitton, G. Ruggeri, A gps-less on-demand mobile sink-assisted data collection in wireless sensor networks, in: Wireless Days (WD), 2014 IFIP, 2014, pp. 1–3, doi:10.1109/WD.2014.7020847.

[41] O. Briante, V. Loscrí, P. Pace, G. Ruggeri, N.R. Zema, Comvivor: an evolutionary communication framework based on survivors' devices reuse, Wireless Pers. Commun. (2015). Accepted for Publication.

[42] C. Bettstetter, G. Resta, P. Santi, The node distribution of the random waypoint mobility model for wireless ad hoc networks, Mobile Comput. IEEE Trans. 2 (3) (2003) 257–269.

[43] J. PUB, National Search and Rescue Manual Volume i: National Search and Rescue System, 1991.

[44] Network Simulator- ns (version 2)(available from http://www.isi.edu/nsnam/ns/).

[45] Fico RR xpress optimization suite, http://www.fico.com/en/products/fico-xpress-optimization-suite/.

[46] B. Milic, M. Malek, NPART - node placement algorithm for realistic topologies in wireless multihop network simulation, in: Proceedings of the 2nd International Conference on Simulation Tools and Techniques, in: Simutools '09, ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), ICST, Brussels, Belgium, Belgium, 2009, pp. 9:1–9:10, doi:10.4108/ICST.SIMUTOOLS2009.5669.

[47] Cisco Aironet 802.11a/b/g Wireless CardBus Adapter, (Data Sheet available on line at. http://www.cisco.com/en/US/prod/collateral/wireless/ps6442/ps4555/ps5818/product_data_sheet09186a00801ebc29.pdf).

[48] E. Natalizio, R. Surace, V. Loscrí, F. Guerriero, T. Melodia, Two families of algorithms to film sport events with flying robots, in: The 10th IEEE International Conference on Mobile Ad-hoc and Sensor Systemsoc Networks and Wireless (MASS), Hangzhou, China, 2013, pp. 319–323, doi:10.1109/MASS.2013.40.

[49] J. He, J. Chen, P. Cheng, X. Cao, Secure time synchronization in wireless sensor networks: A maximum consensus-based approach, IEEE Trans. Parallel Distrib. Syst. 25 (4) (2014) 1055–1065.

[50] J. He, P. Cheng, L. Shi, J. Chen, Sats: Secure average-consensus-based time synchronization in wireless sensor networks, IEEE Trans. Signal Process. 61 (24) (2013) 6387–6400.

Nicola Roberto Zema was born in Italy on 11 January, 1986. He received his BS and MS degree from University “Mediterranea” of Reggio Calabria, Italy in 2009 and 2011 respectively. Since then he has been a PhD Student in University “Mediterranea” of Reggio Calabria while spending several abroad periods in Université de Technologie de Compiègne and Inria Lille. Currently he is a Research Fellow at Université de Technologie de Compiègne, France. His current research activities include Controlled Mobility, Wireless Sensor Networks, Epidemic Networking, Autonomic Networking and Networked Robotic Control.

Enrico Natalizio (Member IEEE) is currently an associate professor with Université de Technologie de Compiègne, France. He obtained his masters degree magna cum laude and his PhD in computer engineering at the University of Calabria in 2000 and 2005 respectively. In 2005-2006 he was a visiting researcher at the BWN (Broadband Wireless Networking) Lab at Georgia Tech in Atlanta, GA, USA. From 2006 till 2010, he was a research fellow at the Titan Lab of the Università della Calabria, Italy. In October 2010, he joined POPS team at Inria Lille Nord Europe, France as a postdoc researcher. His research interest include robot&sensor networks and swarm communications with applications on networking technologies for disaster prevention and management. He is currently an associated editor of Elsevier Ad hoc Networks, and Elsevier Digital Communications and Networks.

Giuseppe Ruggeri, University Mediterranea of Reggio Calabria, Italy (giuseppe.ruggeri@unirc.it), received the MS degree in electronics engineering in 1998 from the University of Catania (Italy). In 2002 he received the PhD in Electronics, Computer Science and Telecommunications engineering from the University of Palermo (Italy). Since 2002 he has been assistant professor in Telecommunications at the Department of Information engineering, Infrastructures, and Sustainable Energy (DIIES) at the University “Mediterranea” of Reggio Calabria (Italy). He authored more than 50 international publications including journals, conferences and book chapters on wireless and mobile systems, protocols, modeling and simulation. He served as TPC member of several leading international conferences (ICC, GLOBECOM, WCNC and others) He served as general co-chair of the SWANSITY 2014 and SWANSITY 2015 workshops. He joined several research projects financed either by Calabria local Government, Italian National Government or European Union. He served as the coordinator of the local research teams at University Mediterranea involved in the projects “Stem”-Net devices to build a Self-Generating access network and DOMUS both financed by the Italian Government. His current interests include the interconnection-integration of heterogeneous wireless network, self-organizing networks, and M2M communication.

Michael Poss is a CNRS researcher at the LIRMM laboratory, a mixed research unit that depends on both the University of Montpellier and the National Center for Scientific Research (CNRS). His current research focuses on robust combinatorial optimization.

Antonella Molinaro (antonella.molinaro@unirc.it) is an associate professor of Telecommunications at the University Mediterranea of Reggio Calabria, Italy. Before, she was with the University of Messina (1998–2001) and the University of Calabria (2001–2004) as an assistant professor; with the Polytechnic of Milano as a research fellow (1997–1998); and with Siemens A.G., Munich, Germany as a CEC fellow in the RACE-II program (1994–1995). She graduated in computer engineering (1991) at the University of Calabria, received a Master diploma in Information Technology from CEFRIEL/Polytechnic of Milano (1992), and a PhD degree in Multimedia Technologies and Communications Systems (1996). Her current research activity focuses on wireless networking, vehicular networks, information-centric networking. She participates in the Information Centric Networking research group (ICNRG) of IRTF and is a member of the NetWorld2020 European Technology Platform. She is in the Editorial board of Computer Networks; Transactions on Emerging Telecommunications Technologies; International J. of Distributed Sensor Networks; EAI Transactions on Internet of Things.