COMVIVOR: An Evolutionary Communication Framework Based on Survivors' Devices Reuse

Orazio Briante, Valeria Loscrì, Pasquale Pace, Giuseppe Ruggeri, Roberto Zema Nicola

Noname manuscript No.
(will be inserted by the editor)

COMVIVOR: an evolutionary communication framework based on survivors' devices reuse

O. Briante · V. Loscri · P. Pace · G. Ruggeri · N. R. Zema

Received: date / Accepted: date

Abstract Mobile devices currently available on the market have a plethora of features and enough computing power to make them, at the same time, information consumers, forwarders and producers. Since they are also provided with a set of sensors and usually battery operating, they are perfect candidates to devise a network infrastructure tailored to function during disruptive events. When everything else fails, they could autonomously reorganize and provide to the civilians and rescue teams valuable services and information. In this paper we adapt and enhance a previous designed framework, capable to epidemically diffuse the proper software updates to its nodes, in order to deploy any kind of service as a prompt response to the needs raised in emergency situations. We further propose and integrate a new smart positioning strategy, to speed up the diffusion of software updates by also keeping under control the overall network overhead. The achieved results show the feasibility of our proposal and how the dynamics of diffusion are enhanced by the smart positioning algorithm.

Keywords Intelligent Positioning · Disaster Recovery · Autonomous Networks

1 Introduction

Large-scale environmental disasters such as earthquakes, floods or terrorist attacks could have a severe impact on the fixed communication infrastructure. Most of the services implemented in it, relying on standard wireless/wired networks, suddenly become non-functional in emergency situations due to both system damage and overuse [14]. On the other side it is very likely that battery-powered wireless personal mobile communication devices will still be up and running after the disruptive event; therefore in this paper we design a framework which allows the reuse of survivors' devices to provide a basic networking communication facility in a post-disaster scenario.

Address(es) of author(s) should be given

Glancing at the plethora of personal mobile devices currently available on the market, we can see that most of them have the potentialities to behave as forwarding nodes [12]; moreover those devices, such as smart-phones, tablets, portable consoles and so on, are widespread, battery powered and, usually equipped with an IEEE 802.11 compliant wireless interface. They could hence be able to communicate even during disruptive events, characterized by power supplying interruptions, to support and restore a possibly damaged communication infrastructure. Furthermore, since those devices are commonly equipped with features as GPSs, microphones, cameras, accelerometers and other sensors, they can, if properly managed, collect information and provide useful services both to survivors and to rescue teams during civil distress periods.

Unfortunately, these devices are often programmed to only behave as *end devices* and, in spite of all their potentialities, they cannot actually communicate in front of a large fault of the traditional communication infrastructures by using their embedded software without making any explicit changes [5]. In particular, the following *ugly truths* represent the main factors limiting the potential of these technologies [5]:

i) Although these commonly used devices can be easily reprogrammed to provide the required services, this operation usually requires the intervention of a skilled operator while the common user is usually technologically agnostic.
ii) To protect the user's privacy, they do not provide any information about the sensed environment to any external entity if not specifically allowed by the user.
iii) In a post-disaster scenario, they could be located in places not easily reachable by rescue teams, for example they can be left in an unsafe building or in a contaminated area.

For all these reasons, the devices are left unable to be exploited to the full spectrum of their capabilities. To further complicate the described context, it is worth to consider that, in a post disaster scenario, the services and the information required by rescue infrastructure can rapidly change in accordance with the evolution of the specific situation. Consequentially, as it could be necessary to continuously adapt the behavior of the network elements, the effective reuse of survivors' devices relies on an effective adaptation and autonomous evolution of such objects to support new services and protocols. These ones can also evolve and autonomously combine with each other to satisfy the needs coming out from the context.

In our previous work reported in [4] we proposed a framework, called *STEM-Net*, which by leveraging an epidemic diffusion model and implementing network virtualization, is capable to epidemically diffuse entire protocol stacks to a set of nodes. As a natural consequence, in this paper we propose to apply *STEM-Net* to reprogram survivors' devices by making them able to autonomously organize and form or restore a communication infrastructure to help people during emergency situations or environmental distress.

According to our proposal, when a disaster happens, and all traditional communication services are jeopardized, an user can just enable the *STEM-Net* framework on his/her phone so that this latter can *reconfigure* itself by learning or acquiring new *software updates* which implement the new *applications* and/or *communication protocols* required by the situation. These *software updates* will be referred in the following as *evolution modules*.

In our view *evolution modules* are at first provided by an autonomous flying drone which is delivered on the disaster area by the rescue teams. In a second stage *evolution modules* are passed from device to device according to an epidemic paradigm. By adapting to the changing needs of situations and rescue teams, the *evolution modules* implement the needed new services and protocols and, by leveraging on single nodes' self-evolution, the performances of the network as a whole are also improved.

A second contribution of this paper comes from the consideration that the diffusion rate of an epidemic (*i.e.,* the *evolution modules* in our case) strongly depends on the place the first infection take place [18]. Hence, we also propose a new protocol, based on a smart positioning strategy, to effectively drive the flying drone deployed by the rescue teams with the aim of improving the *STEM-Net* performance, in terms of both increase in the diffusion speed of the *evolution modules* and reduction of the overall network overhead.

This enhanced communication framework, obtained through the combination of an improved *STEM-Net* framework with a wise drone driving protocol, has been named `COMVIVOR` (**Com**munication for Sur**vivor**s) in order to highlight the specific aim of such new approach.

We simulated the behavior of `COMVIVOR` throughout the Network Simulator 2 (ns-2) [1] by evaluating its performance in a realistic wireless simulation environment. The obtained results testify the effectiveness of our framework in performing the self-evolving process of network nodes; moreover, we also show how the performance of the whole system can be greatly improved by introducing a smart positioning technique originated by teorethical considerations of the epidemic model.

The rest of the paper is structured as follows. Section 2 summarises recent works in which autonomic devices reconfiguration and virtual network deployments have been used in emergency and disaster scenarios. Section 3 describes the epidemic spreading process within the `COMVIVOR` protocol whilst the new autonomic positioning strategy, based on a smart positioning strategy, is described in Section 4. The simulation campaigns and the obtained results are presented and discussed in Section 5; finally, the Section 6 concludes the paper presenting few future research directions and open issues.

2 Related Work

In the last years, ICT-related disaster monitoring and management have attracted many researches because of the huge impact on the modern society of a disastrous event that affects large numbers of people and cause loss of

lives and infrastructures. In this section we briefly survey the literature for *i)* disaster monitoring and management frameworks and, *ii)* autonomic devices reconfiguration.

2.0.1 Disaster Monitoring and Management Approaches

Many frameworks and network strategies have thus been proposed and deeply investigated to address the issues related to a disaster scenario, from different points of view [7, 13, 15, 17, 22–26]. Specifically, the contribution of [7] consists in a general discovery protocol that is asynchronous. The authors verified its effectiveness both analytically and through real-world experiments on smartphones. In [17], authors envisage the network auto-configuration software (NAS), as one of the fundamental requirements in the context of Discovery Access Network (DRAN) and they propose a Tree-based multihop Disaster Recovery Access Network (TDRAN). In [22], authors propose a different mobility model to characterize and capture specific features of the topology after a disaster. This re-definition of the topology will be helpful for the development of specific routing approaches in the context of Disaster Recovery system [13]. In [25] authors present concepts really close to our vision of the ICT role in the disaster recovery treatment. They propose a resilient network architecture based on resource units that are movable and deployable. Though their approach embraces, in some way, our phylosophy, we do not introduce specific units, but we leverage on user'mobile devices, by re-defining their role. Moreover, we consider and describe a better positioning of the *plague-spreader* node. In [23], authors show the feasibility of using a smartphone as relay for communication purpose and the advancements of user-driven networking powered by communication devices independent of operator networks.

In [24], authors propose a framework for disaster management by implementing ad hoc communication in three different scenarios, adaptively adjusting the transmission power to reduce the overall network energy consumption. However, they did not consider any mobility model to validate their analysis and they also did not address the issues related to the communication between devices supporting different interfaces and standards.

In [26], authors propose a mobility model that includes the impact of the disaster on the transportation network by modeling the population and relief vehicle movement. The mobility model exploits a general communication scenario based on the DTN (Delay Tolerant Network) paradigm. They consider network devices that exchange buffered messages when they come into communication range. In addition, an opportunistic forwarding paradigm is used in [15] to investigate the potentials of novel routing methods applied to emergency scenarios. The authors evaluate the overall performance by taking into account parameters like the number of people involved or the number of victims, in order to measure their impact on the performance of routing methods.

2.0.2 Autonomic Devices Reconfiguration

Recently, the idea of using smart-phones to support the network infrastructure damaged by the disaster began to take hold as shown in [8]. In this work the authors propose a smart-phone/PDA based disaster management system working on peer to peer communication only and supporting disconnected operation. By assuming the absence of a network infrastructure, the proposed approach tried to incorporate the notion of opportunistic network by using peer-to-peer communication between relief workers.

In contrast to these proposals, our approach tries to be more proactive because it supports the discovery of survivor devices and makes possible the implementation of behavioral changes by allowing the evolution of network devices through the reception of the *evolving modules*. Furthermore, we consider and evaluate the possibility to opportunistically move/position a mobile device which is in charge to provide the *evolution modules* required by the situation.

3 COMVIVOR: The epidemic spreading process

Evolution modules are provided to mobile devices in the disaster area by means of an *epidemic process*. The *epidemic process* is triggered and sustained by special purpose nodes which: *i)* advertise *evolution modules*, and *ii)* provide long range connectivity, for example through a satellite link. In our vision those special nodes can also autonomously roam through the disaster area and in the following they will be referred as *plague-spreaders.* In a real case scenario, the *plague-spreaders* can be flying drones which go ahead of the rescue teams to collect the first information about the disaster area, create basic networking and monitoring facilities and pave the way for the later human intervention. The possibility of using such flying devices as an effective communication station is displayed in [11] [21] [27] and [28].

We further assume that a consistent part of the end devices implements a COMVIVOR demon, this demon remains silent during all the normal life of a device but wakes up only when the user explicitly activate it in front of the disaster. Let us note that we expect that the effort to activate COMVIVOR demon will be comparable to the the effort required to run any other application on the device, hence we guess that it will be affordable by common users with no specific skills.

When the COMVIVOR demon is activated it looks on all the short range interfaces, like IEEE 802.11 or Bluetooth, for any incoming signal from a *plague-spreader*. In order to prevent any malicious use of COMVIVOR all the messages from a *plague-spreader* are signed with a secret key owned only by national security. A node accept an evolution module only after it has verified its signature by using a public key which is periodically published by government agencies.

Once a unit has received a message from a *plague-spreader* it goes through the infection process where the *evolving module* is passed from node to node like a virus going from sick beings to healthy ones.

Our current work takes its basic functionalities from our previous research on epidemic diffusion [4] where a fixed mobile station diffuses evolution modules. Since our present research focuses on the effectiveness of a better positioning of the *plague-spreader* node, herein we briefly remind the basic working principles of the original epidemic spreading process.

In the following, the dissemination process of a single *evolving module* is presented. A more general case, in which different modules are simultaneously distributed, can be accomplished by following the same procedures.

3.1 Basics

As assessed in [4], in order to disseminate an evolving module in the network, three concurrent processes are carried out: *(i)* the *dissemination process*, in which the spreading of information about new evolving modules takes place; *(ii)* the *individual decision process*, in which each node individually decides whether to accept or not an evolving module; *(iii)* the *infection process*, in which the executable implementation of an evolving module is transmitted to the requesting nodes.

Three types of node are defined in our framework: *plague-spreader* nodes, *susceptible* nodes and, *infected* nodes.

Mobile *Plague-spreader* nodes, now introduced, initially advertise the availability of an evolution module through the broadcasting of so-called *DNA-packets* and provide, on demand, an executable implementation of it.

Susceptible nodes are potential recipient of an evolving module, which can be accepted with a given degree of susceptibility. In our framework the set of nodes *susceptible* to an evolution module is represented by all those nodes where: *(i)* the `COMVIVOR` demon has been activated and *(ii)* the evolution module has not been acquired yet.

Infected nodes are the ones that have acquired the evolving module and will contribute to advertise and spread it.

3.2 Dissemination process

The dissemination process is accomplished through the periodic broadcasting of *DNA-packets* that are generated by *plague-spreader* and *infected nodes* and processed by *susceptible* ones. To keep constant the signaling overhead, the emission rate of *DNA-packets* is dynamically adjusted so to remain constant in the network. Specifically, if a node senses that other nodes are broadcasting the same *DNA-packet*, then it reduces its transmission rate accordingly. A *DNA-packet* carries the description of the advertised evolving module that contains the identifier (ID) of the *plague-spreader*, the ID of the module, the type (*i.e.,*

Protocol Stack or Service), and a description of the resources required to run the module, in terms of memory and computational load.

By analyzing the received *DNA-packets*, a node can infer other information as well as an estimation of its capability to run it. For instance, a node may estimate how popular is an evolving module by considering how many neighbors (*plague-spreaders* and *infected* nodes) are advertising it. Similarly, a node may estimate the distance from the source of a DNA-packet by considering the received signal strength.

3.3 Individual decision process

Although many *evolution modules* may be advertised in the network, not all of them could be suitable to any node. Thus, when a node receives a *DNA-packet* goes through a *decision process*. This process considers several input parameters related to the node and the network conditions (*e.g.,* hardware characteristics of the node, energy constraints, number of neighbors). As a result of the decision process, an evolving module can be stored, executed, or ignored. Analogously, a module already active may be switched off and discarded due to a change in the available node resources or in the network conditions. The devising of a suitable *decision process*, as a *stimulus-response* mechanism, has been addressed in our previous works reported in [4, 9]. We refer the readers to those works for further detail on the *decision process*. For the purpose of this work we model the *decision process* by giving: *(i)* the susceptibility F of the node toward a specific module, and *(ii)* the possible *switch off and discarding event* G.

In more detail, F represents the probability for a given node to accept an *evolution module* that has been advertised in a *DNA-packet*, while G represents the average time after which a node is forced to discard an *evolution module*. It is worth to be noted that in a post-disaster scenario low values for G are expected due to: i) the high level of network disruption and ii) the impossibility to recharge the nodes which implies a short lifetime of latters.

3.4 Infection process

Through the infection process, the evolving module propagates from the *plague-spreader* to its one-hop neighbors, and then from the latter (if infected) to their one-hop neighbors thus following an epidemic diffusion. Therefore, hop-by-hop, the evolving module can reach all the nodes belonging to connected subsets in the network.

Whenever a *susceptible* node accepts a new evolving module, it starts the infection process; in particular, each implementation of an evolving module consists of an executable code divided into a number of elementary units called *chunks* that can be transmitted on the broadcast medium without any further fragmentation. `COMVIVOR` assumes that the number of chunks composing the

executable code can vary from module to module and it is advertised by the *DNA-packet*. The download of the entire code is performed through the broadcasting of two packet types, the *Interest* and the *Data*. An *Interest* is used to request a specific set of chunks and includes the identifiers of both evolving module and the chunk set. When the *plague-spreader* (or any *infected* node) receives an *Interest*, it broadcasts the corresponding set of chunks within data packets. The *Interest-Data* exchange continues until the entire code is downloaded. In case of a packet loss, the *Interest* is transmitted again but the requested set of chunks is modified according to new searched data. In order to reduce the collision probability arising from multiple simultaneous transmissions (*e.g.,* in the presence of two or more *infected* nodes that can reply to the same *Interest*), each node i waits a random defer time T_D before transmitting the *Data*. During T_D, i monitors the channel to detect the same transmission performed by other nodes. Finally, if anyone has issued the *Data*, it sends the packet.

4 COMVIVOR: The Spreader Autonomic Positioning

The application of a smart positioning technique, can be really advantageous in several and different network scenarios [2, 3, 6, 19, 20]. In the COMVIVOR context, it is intended as the capability of the *plague-spreader* node to move to a different position according to a specific behavior or goal to be reached like, in our case, an improvement on the diffusion performances of the modules. To accomplish this we have devised a mechanism based on selective broadcast of node density information in *probe* packets in such a way that, after a *probe phase* that can be reiterated in case of topology changes, every node can know the value of the *highest* node density point of its network cluster area. By the knowledge of this point a spreader node can directly move to this position prior to any other action. Starting the infection from a denser area highly enhances the speed of infection as our results show. In our work the density perceived by a node is intended as the number of reachable neighbors versus the node's transmission range. If we consider that all the nodes are of the same kind, we can just use the total number of neighbors as an indication of the density of a circular area of which the sampling node is the center.

A different version of the algorithm has been devised in order to handle extreme cases in which the density value is not unique but it is uniform along the simulated area such as in an equispaced grid topology. In this case the spreader node is moved by considering a directional criterion: in particular, the final position is the one that maximize the directional dispersion of an epidemic (*i.e.,* the barycenter of an equispaced lattice of points). According to the relation between *highest density* value and the estimated general density of the whole network area, the algorithm itself can choose the more convenient method to apply in order to speed up the infection process comparing the highest density value with the general area density and using the barycenter method only in the equispaced cases.

Table 1: Probe Packet composition.

Local Values	Bytes	Best Values	Bytes	Top Values	Bytes	Bottom Values	Bytes
Position X	8	Position X	8	Position X	8	Position X	8
Position Y	8	Position Y	8	Position Y	8	Position Y	8
Perceived Density	2	Highest Density	2				

4.1 Topological values diffusion

In our work we suppose that a mobile flying device (*i.e.,* Drone, UAV, etc...) is already present on the field as there are no long-range communication for the reasons illustrated beforehand. In an initial phase, every node can broadcast a probe packet containing the sender ID and ten fields as shown in table 1: the sender ID coordinates, its perceived density, the coordinates of the node that perceived the *highest density* the broadcaster node ever had information about and the actual value of perceived density associated with the latter; to those are added the coordinates of top and bottom points of the disaster area in order to calculate the barycenter as discussed in the previous section. We have used 8-bytes floating point numbers for the coordinate implementation and 2-bytes integers for the density values and IDs. In this way we model the possibility to use real GPS coordinates and a relatively large number of nodes [10]. In the very first broadcast phase, the only nonzero fields of the packet a node sent are its own coordinates and ID; once another node receive the probe packet, it can acquire the information about the actual presence of a neighbor in his receiving range. At the end of the first broadcast phase (after a time probe $= Tp$), every node can have a first overview of the nodes in its surrounding (*i.e.,* Perceived Nodes - PNs) and it can calculate its own initial perceived density to be broadcasted by using new probe packets.

At the second broadcast run, every node will compare the received *highest density* value with the local and if found higher they will start to broadcast that as their *highest density* heard. As the broadcast process goes on, when a node receive a *foreign* density value that is higher than the local, the new value is stored and broadcasted in the new probe packet. At the end of the broadcast runs, every node connected to any subset of the total nodes in the network, will know the *highest density* value of that subset coupled with the coordinates of the related node.

This broadcasting information scheme needs a maximum time $\triangle T_{MAX}$ in order to guarantee the perfect knowledge of the denser area to all the nodes participating to the broadcasting process. This maximum time has been estimated by considering the worst case analysis applied to a standard IEEE 802.11g multi-hop wireless network as described in appendix 7.

According to this analysis, when a spreader node appears in the network it can know, after a quite reasonable time, the position where it should move to maximize the number of nodes to infect.

In the barycenter version, the same principles are applied for the propagation of the top and bottom values for every cluster involved. In this version

only the highest or lowest values of the coordinates are propagated to any reachable node.

4.2 Spreader Node Movement

The density information can be updated periodically by taking into account the lower bound due to the natural delay of the propagation scheme (*i.e.,* see the computation of the $\triangle T_{MAX}$ value) and also differentiating the nodes that are already mutated from the ones that still need to be reached. In this fashion, once the area with maximum density has been discovered, the spreader node can directly move to the densest area, starting there the mutation process. As our result will show, the speed of mutation is dramatically increased just from moving the node only one time.

Of course after the individual mutation process, each node can be automatically excluded by the density calculation algorithm; in this way, when the density discovery algorithm starts again, the spreader node will be eventually reached by the updated information about the densest area of un-mutated nodes. Since, the density information exchanges are faster than the mutation process of the nodes, when they are activated both, the former can feed up information to the spreader while the infection is ongoing. This feature shows its importance mostly when the number of nodes in a network is not fixed. In the disaster recovery scenarios it is quite likely that large subsets of nodes deactivate temporarely while another subset joins the network. In this case it is necessary to swiftly update the information about the nodes distribution in reaction to these events.

In algorithm 1 is shown a pseudo-code description of the proposed topological diffusion strategy based on a selective broadcast information mechanism. The pseudo-code has been differentiated to describe both the spreader and the generic node behaviors.

After a certain number of broadcast we have observed that the values converge within the cluster the nodes belong to.

Dedicated simulation shows that the algorithm can deliver information about the presence of nodes agglomeration point in a bounded number of iterations (see fig. 1). In these preliminary simulations we used a grid scenario of 100 equispaced nodes with the distance among them roughly equal to their transmission range, their characteristics equal to the ones we used in next simulation by running `COMVIVOR` meta-protocol. We put a small aggregation point consisting of a set of nodes placed in a corner, effectively doubling the local density, and a *plague spreader* node in different positions: in the opposite corner (16 hops far), in the center of the lattice (8 hops far) and in an adjacent corner (9 hops far). From the figure we can observe that the needed number of iteration required to compute an estimation of the center of the agglomeration point to the monitoring node, is roughly dependent on the distance in hops from them. Specifically the same number of iterations as hop count is needed for the first information to arrive (un-tuned to the actual point of maximum

Algorithm 1 Density Discovery and Diffusion: Node and Spreader behaviors

⇒ i^{th}**Node Behavior**:
while $T <= endtime$ **do**
  while $K * Tp \leq T \leq (K+1) * Tp$ **do**
    $Collect\ Probes\ from\ Perceived\ Nodes\ PN$;
    if $T = (K+1) * Tp$ **then**
      for $j = 1$ *to* PN **do**
        $Calculate\ distance\ (i, j)$;
      $evaluate\ MaxDistance() = Md$;
      $perceivedDensity = PN/(Md * \pi^2)$
      if $perceivedDensity > HighestDensity$ **then**
        $HighestDensity = localData$;
      $Send\ Probe()$;
      $K++$;

⇒**Spreader Node Behavior**:
$Every\ \triangle T_{MAX}$;
if $Maximum\ Density\ area\ detected$ **then**
  Move to **denser area**
else
  Move to **Barycenter**

density) while it is necessary to add a number of iterations roughly equal to twice the diameter of the epidemic in order to deliver a good estimation of the epidemic center (*i.e.,* the point of highest node density).

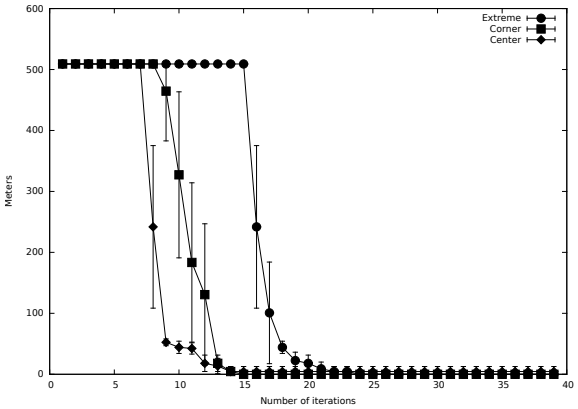


Fig. 1: Error in computed Epidemic Position vs. Number of Iterations.

5 Positioning algorithm: simulations and results

To evaluate the performance of `COMVIVOR` we implemented the proposed framework in ns-2 [1] and considered the scenario in Fig. 2, which represents an

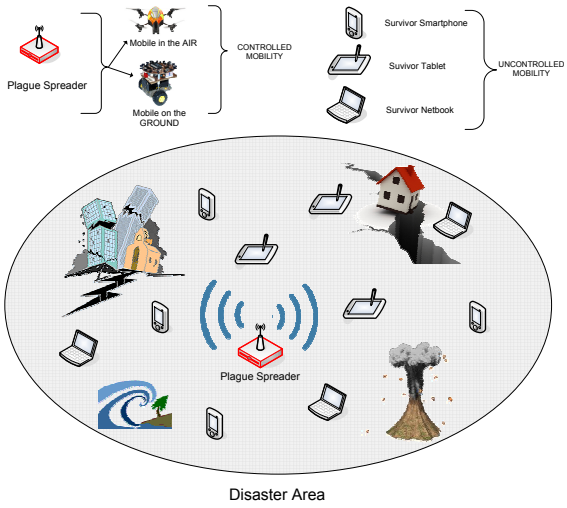


Fig. 2: General simulation scenario.

Table 2: 802.11g Simulation Parameters.

PHY Parameter	Value		MAC Parameter	Value
Frequency	2.4 GHz		SlotTime	9 μs
Receive Sensitivity	-86 dBm		SIFS	16 μs
Transmission Power	18 dBm		Preamble Length	96 bit

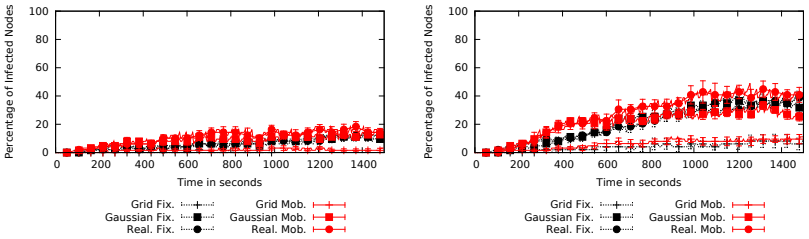
urban area as a $500m$-wide square grid. For sake of simplicity, at this stage we considered all nodes equipped with a single IEEE 802.11 physical interface. A Ricean fading model accounts for multi-path effects due to various obstacles, buildings and trees. We further assumed that the executable code of the evolving module is $1MB$ long. Regarding the fragmentation, `COMVIVOR` assumes each chunk is 1024 bytes long. For the simulation purposes, we translated the parameters F and G into floating point numbers representing the probabilities of accepting an *evolving module* and the averaged time to trigger a discarding event.

In our previous work [4] we already evaluated the effectiveness of the diffusion mechanism and we proceed to assess its capabilities in presence of an opportunistic smart positioning of the *plague-spreader* node.

As shown in Table 2, Physical and MAC parameters are based on IEEE 802.11g and transmission power and receiver sensitivity figures are taken from data-sheets of devices available on the market.

In our set of simulation we tested the system by placing the wireless nodes within the simulation area according to the specific scenarios shown in Fig. 4: *i)* equispaced grid, *ii)* Gaussian distribution, *iii)* realistic distribution. In particular, the Gaussian scenario has been generated by following a Normal distribution criterion with the aim of creating denser areas, even if the full

connectivity between all the nodes of the networks is not guaranteed. On the contrary, the realistic topology has been generated by using a specific framework called NPART [16], capable of designing network topologies whose statistical characteristics are similar to the ones measured in real networks. All the simulated scenarios have been designed to support a fixed density value of $6.5 * 10^{-3} nodes/m^2$, an expected $E\{G\} \in [150, +\infty[$ seconds and we fixed the value of susceptibility $F = 0.5$. The value $E\{G\} \to +\infty$ corresponds to the case when *infected* nodes never discard the evolving module while the lower values try to reproduce the high network disruptive environment where nodes could suddenly cease to function and others may replace them in the geographical vicinity. We analyzed the time evolution of the infection by evaluating the number of infected nodes over the total of nodes. Each value is the result of a 20 simulative runs which outputs were processed with 95% confidence intervals and shown where applicable. Initially the *plague-spreader* node is located at the center of the network area for the *Gaussian* topology and in a randomly selected corner for the others that vary within runs.

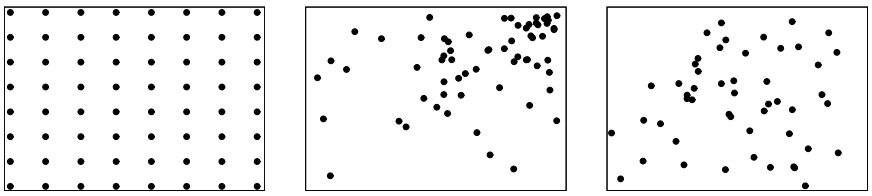


(a) $E\{G\} = 90s$ (b) $E\{G\} = 170s$

(c) $E\{G\} = 570s$ (d) $E\{G\} = \infty$

Fig. 3: Fraction of *infected* nodes varying time and topology. $6.05 \cdot 10^{-3}\, Nodes/m^2$; $F = 0.5$.

Our main objective is to evaluate if `COMVIVOR` allows to obtain a steady state in the epidemic diffusion process, and how fast it is reached. It is worth to be noted that the target of the spreading algorithm is to reach a given fraction of the nodes, which is not necessary the totality of them. This could be the case of an applicative service which requires that only a small fraction



a) b) c)

Fig. 4: Simulation scenarios $6.05 \cdot 10^{-3}\, Nodes/m^2$: a) Grid scenario, b) Gaussian Scenario, c) Realistic Scenario

of nodes execute the proper evolution module. In Figure 3 it is shown the cumulative curve of infected nodes versus time with different $E\{G\}$ and grouped by topology. Regardless the considered scenario, the proposed mobility framework achieves better performances in terms of mutation dynamics by allowing to reach the steady state in much less time. From figures 3c and 3d it can be shown that, by using the mobility framework the system always reaches the steady state in less time rather than when it uses the fixed positioning system. However when $E\{G\}$ is low, as shown in 3a and 3b, the whole system has difficulties in reaching a fairly reasonable number of mutated nodes regardless of a better positioning of the *plague-spreader* node. This means that the system has difficulties in the *evolving module* delivery when the average lifetime of a node is comparable to the time needed to a module transfer. However, this possibility for the $E\{G\}$ value is quite uncommon. For example in Figure 3a the network does not even reach 20% of mutated nodes due to an $E\{G\}$ value of $90s$. The scenario in which we noticed the lowest system performance is the grid topology one, where the overhearing of data packets is limited by the equispaced dispersion of the nodes by making the mutation process more time consuming and prone to disruption in the cases of low average lifetime. However, the smart positioning of the *plague-spreader* node allows to improve the system performance also in this unfavourable scenario. To better clarify the obtained results, we also shown the cumulative curve of infected nodes, grouped by the $E\{G\}$ values, in order to make similar considerations. Figure 6 shows that, starting the mutation process from the previously computed network barycenter, it is possible to greatly improve the speed of mutation. Figure 5 shows instead the results varying $E\{G\}$ with the Gaussian topology. The curves crossings are due to the fact that the biggest percentage of the whole network nodes lies in the region where the mobile node moves to and, even if this percentage is infected first, there is still a sensible number of isolated peripheral nodes. Those last, are reached first if the node remains nearer to the network periphery rather than moving to the point of maximum agglomeration. Finally figure 7, related to a Realistic NPART-generated topology, shows a general improvement using the mobility framework in all of the previously introduced aspects.

Furthermore, in order to quantify the impact of the mobility framework on the system overhead, we also measured the so called *mutation overhead* representing the number of bytes transmitted by the nodes with the specific purpose of epidemic diffusion, averaged by the number of mutations. The obtained results are shown in figure 8 in which the mutation overhead is consistent with the topology and mobility scenarios while it varies with the average node lifetime. In fact, the increase of this parameter causes less mutations, less congestion and less bytes sent as a consequence.

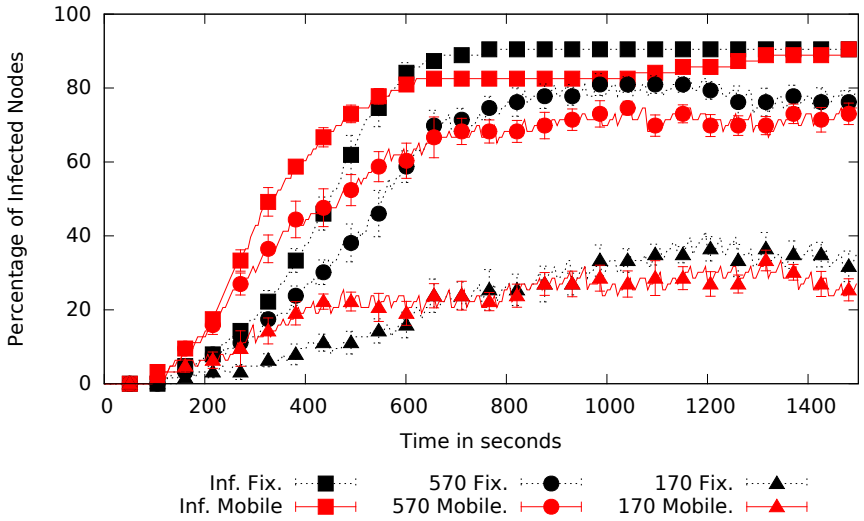


Fig. 5: Fraction of *infected* nodes varying time and $E\{G\}$. $6.05 \cdot 10^{-3}\, Nodes/m^2$; $F = 0.5$. Gaussian Topology.

Finally, the figure 9 shows the peak value of mutated nodes by varying: *i)* network scenario, *ii)* presence of mobility framework and *iii)* discarding time $E\{G\}$. As expected, the peak value of mutated nodes is exactly equal to the amount of nodes within the network when the lifetime if infinite and the network connectivity is always guaranteed (*i.e.,* Grid and Realistic scenarios); however, as the node lifetime decreases, the peak value of mutated nodes will decrease concordantly. Starting from the grid scenario, where by definition there are no high concentration points, the lower lifetime greatly decreases the peak number of mutated nodes without the diffusion enhancement due to the positioning system. In the other case, by using the mobility framework, the spreader node is capable to place itself where it can maximize an omni-directional diffusion and compensate for the network disruption. The same concept can be applied to the other scenarios where the presence of mobility framework enhance the infection diffusion as it keeps the most dense area

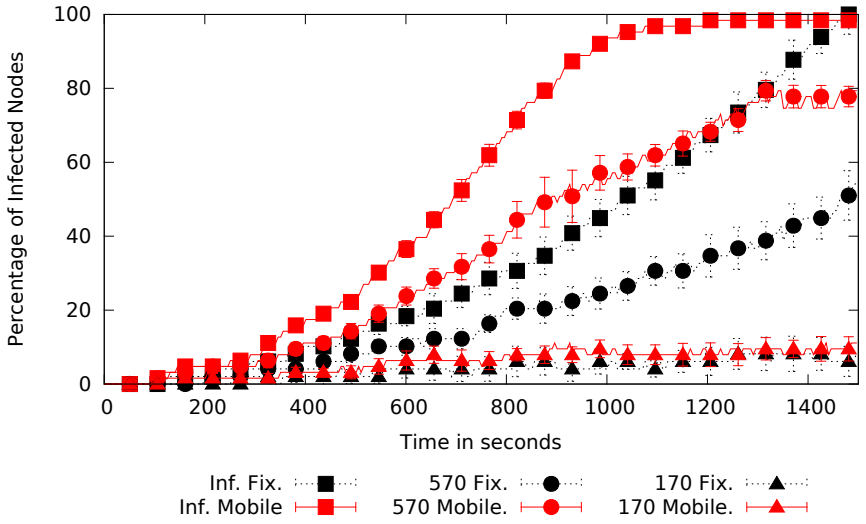


Fig. 6: Fraction of *infected* nodes varying time and $E\{G\}$. $6.05 \cdot 10^{-3}\, Nodes/m^2$; $F = 0.5$. Grid Topology.

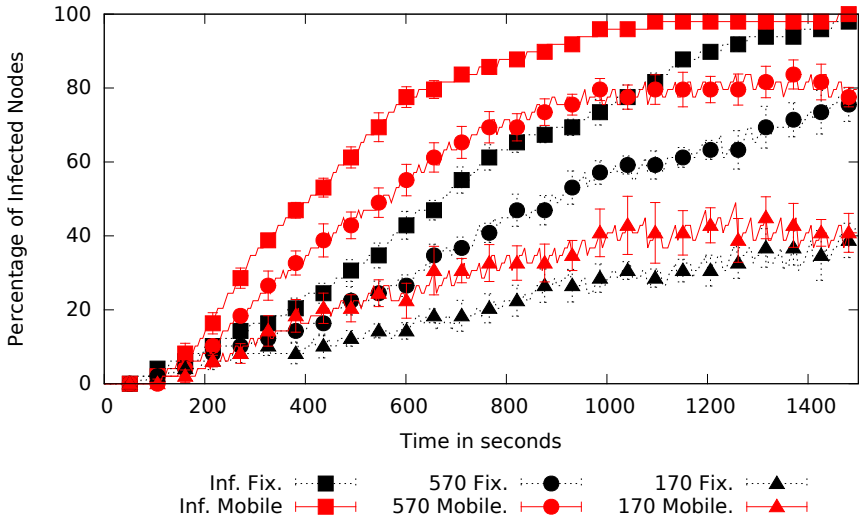


Fig. 7: Fraction of *infected* nodes varying time and $E\{G\}$. $6.05 \cdot 10^{-3}\, Nodes/m^2$; $F = 0.5$. Realistic Topology.

of nodes with readily available software modules and constantly spreads the infection to the nodes that have recently discarded the *evolving module*.

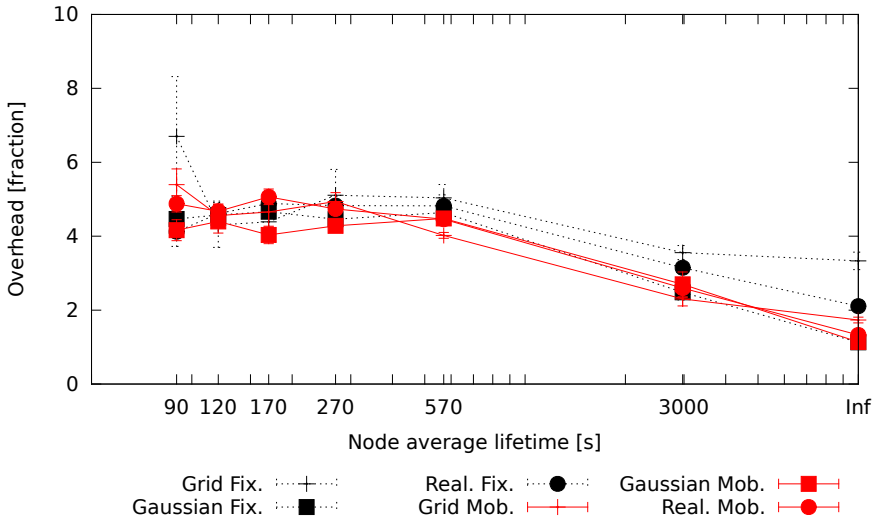


Fig. 8: Mutation Overhead.

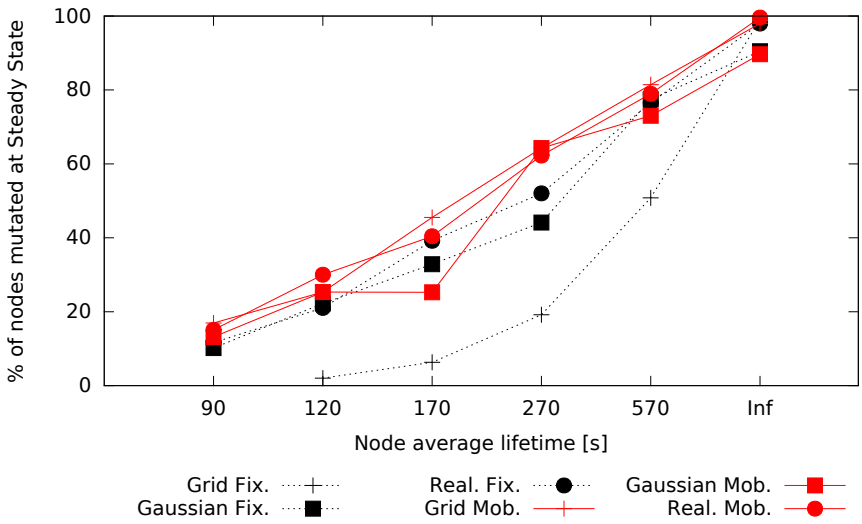


Fig. 9: Peak value of mutated nodes.

6 Conclusion

In this paper we designed and tested COMVIVOR, a communication framework suited for managing, in an effective way, emergency and disaster scenarios. Leveraging on concepts like virtual networking, smart positioning strategy and epidemic networking, the proposed framework takes advantages from the

reuse of the users devices in situations of public distress. In particular, it enables the devices scattered in the field to autonomously cooperate during large scale environmental disaster and ease the operations of the rescue teams by providing a solution to restore connectivity and load on-demand services and protocols. We have shown the improvements that the an opportunistic and smart positioning can add to the whole framework by the proposal of an algorithm for autonomous network analysis linked to a smart mobility scheme applied to flying mobile devices. We have demonstrated that, providing the values of local network density to a mobile entity in order to make it able to start the diffusion of evolving modules from the most dense area, would greatly improve the speed and maintenance of the infection even in highly disruptive conditions.
As future works, we plan to continue our research paths in multiple ways. For instance, even if the system is already capable to handle multiple *plague-spreaders*, it should be devised an active coordination system between them coupled with a careful tuning of the network parameters. It should also be analyzed the case of multiple concurring infection and a decision engine that is capable to discern between multiple module options in a comprehensive manner. Moreover, it would be very interesting the evaluation of the controlled mobility paradigm on the effectiveness of the epidemic propagation speed. In practice, the controlled mobility approach would allow the *plague-spreader* to move again by following the same logic, once the region where it previously moved has been infected.

7 APPENDIX A: Density discovery delay

In this section we provide an analytical estimation of the maximum delay time in order to acquire the perfect knowledge of the densest area to all the participating nodes. The proposed *worst case analysis* has the aim of demonstrating that, even in presence of a high amount of collisions and quite old and slow transmission technology such as IEEE 802.11b (Table 3), the density discovery mechanism proposed in this paper and implemented by using a selective broadcasting mechanism, can be executed in a fair amount of time.

Let T_{header} be the time spent in all the protocol headers on each PROBE packet including the PLCP and 802.11 headers. We consider that PLCP header is transmitted at 1 Mbps while the others are transmitted at R Mbps (*e.g.* 11 Mbps).

$$\begin{aligned} T_{header} &= T_{PLCP} + T_{802.11} \\ &= 192 + 28 * 8/R \\ &= 192 + 224/R \end{aligned}$$

By considering a data rate of 11 Mbps we have $T_{header} \cong 213\mu s$

The time spent by each probe packet T_{PROBE} for one single hop is the sum of DIFS period, the station's back-off computed for the worst case, the headers transmission, the link-layer ACK and the SIFS period.

Table 3: IEEE 802.11b relevant parameters

Parameter	**Value**
Channel rate	11Mbps
SIFS	10 μs
DIFS	50 μs
Slot Time T_{slot}	20 μs
$PLCP_{Header}$	192 μs
$[CW_{min}, CW_{max}]$	[32,1024]
Probe packet	1500 byte
Ack packet	14 byte

$$\begin{aligned} T_{PROBE} &= DIFS + T_{Boff} + T_{header} + T_{L-PROBE} + \\ &\quad + SIFS + T_{PLCP} + T_{L-ACK} \\ &= 50\mu s + T_{Boff} + 213\mu s + (1500 * 8/R) + \\ &\quad + 10\mu s + 192\mu s + (14 * 8/R) \end{aligned}$$

In particular the T_{Boff} value, representing the most dominant terms, is computed according to a worst case analysis in which we suppose there are six consecutive collisions at MAC layer before to gain a valid transmission and the congestion window dimension CW_{min}, is doubled at each collision as shown in the following formula:

$$T_{Boff} = [\sum_{i=0}^{5} 2^i * CW_{min} * T_{slot}] + [CW_{max} * T_{slot}]$$

Due to the back-off timer mechanism, the slot duration is not always a fixed value but it differs depending by the specific status (i.e., idle, occupied, successful transmission, unsuccessful transmission due to frame error or collisions). By following the worst case analysis, we always considered the greatest value for the slot duration T_{SW} represented by the average time the channel is sensed busy by each station because of a successful transmission:

$$\begin{aligned} T_{SW} &= T_{header} + T_{L-PROBE} + SIFS + T_{L-ACK} + DIFS \\ &= 213\mu s + 1091\mu s + 10\mu s + 11\mu s + 50\mu s = 1375\mu s \end{aligned}$$

Thus the maximum T_{Boff} value can be computed as follow:

$$T_{Boff} = [\sum_{i=0}^{5} 2^i * CW_{min} * T_{SW}] + [CW_{max} * T_{SW}] = 4.18s$$

furthermore, we can argue that the time for sending each probe packet towards a single hop transmission, is slightly bigger than the back-off collision time:

$$T_{PROBE} = T_{Boff} + \varepsilon$$

In conclusion, to compute the ΔT_{MAX} value to propagate the broadcasting information in a multi-hop path of N hops and to receive back the related ack message, we can use the following relation:

$$\Delta T_{MAX} = 2 * N_{hops} * T_{PROBE}$$

In particular, by considering a network 7 hops long, it is possible to complete the broadcast process, to be aware of the most dense location area, in less than one minute (*i.e.* about $58.52s$).

Acknowledgements This work has been carried out under the framework of STEM-Net, PRIN-National Italian Project # H21J11000050001, financed by the Italian Ministry of University and Research.
The research of Nicola Roberto Zema is partially supported by European Union (EU), European Social Fund (ESF) and Calabria Local Goverment. This paper reflects the views only of the authors, and the EU, the ESF and Calabria Local Government cannot be held responsible for any use which may be made of the information contained therein.

References

1. Network Simulator- ns (version 2). available from http://www.isi.edu/nsnam/ns/
2. Akkaya, K., Senel, F., Thimmapuram, A., Uludag, S.: Distributed recovery from network partitioning in movable sensor/actor networks via controlled mobility. Computers, IEEE Transactions on **59**(2), 258–271 (2010). DOI 10.1109/TC.2009.120
3. Akyildiz, I., Xie, J., Mohanty, S.: A survey of mobility management in next-generation all-ip-based wireless systems. Wireless Communications, IEEE **11**(4), 16–28 (2004). DOI 10.1109/MWC.2004.1325888
4. Aloi, G., Bedogni, L., Felice, M.D., Loscrì, V., Antonella, A.M., Natalizio, E., Pace, P., Ruggeri, G., Trotta, A., Zema, N.R.: Stem-net: an evolutionary network architecture for smart and sustainable cities. Transactions on Emerging Telecommunications Technologies **25**(1), 21–40 (2014). DOI 10.1002/ett.2785. URL `http://dx.doi.org/10.1002/ett.2785`
5. Aloi, G., Felice, M.D., Loscrí, V., Pace, P., Ruggeri, G.: Spontaneous Smartphone Networks As User Centric Solution for the Future Internet. IEEE Communication Magazine (In Press)
6. Aziz, A., Sekercioglu, Y., Fitzpatrick, P., Ivanovich, M.: A survey on distributed topology control techniques for extending the lifetime of battery powered wireless sensor networks. Communications Surveys Tutorials, IEEE **15**(1), 121–144 (2013). DOI 10.1109/SURV.2012.031612.00124
7. Bakht, M., Trower, M., Kravets, R.H.: Searchlight: Won't you be my neighbor? In: Proceedings of the 18th Annual International Conference on Mobile Computing and Networking, Mobicom '12, pp. 185–196. ACM, New York, NY, USA (2012). DOI 10.1145/2348543.2348568. URL `http://doi.acm.org/10.1145/2348543.2348568`
8. Deb, D.D., Bose, S., Bandyopadhyay, S.: Coordinating disaster relief operations using smart phone / pda based peer-to-peer communication. International Journal of Wireless and Mobile Networks (IJWMN) **4**(6) (2012)

9. Di Felice, M., Bedogni, L., Trotta, A., Bononi, L., Panzieri, F., Ruggeri, G., Aloi, G., Loscri, V., Pace, P.: Smartphones like stem cells: cooperation and evolution for emergency communication in post-disaster scenarios. In: Communications and Networking (BlackSeaCom), 2013 First International Black Sea Conference on, pp. 28–33. IEEE (2013)
10. Garmin International, I.: GPS 16x Technical Specifications, [Page 27]: Appendix B: Garmin Binary Output Format; Position Record. Garmin International, Inc.
11. Giorgetti, A., Lucchi, M., Chiani, M., Win, M.Z.: Throughput per pass for data aggregation from a wireless sensor network via a uav. Aerospace and Electronic Systems, IEEE Transactions on **47**(4), 2610–2626 (2011)
12. Iera, A., Molinaro, A., Paratore, S., Ruggeri, G., A.Zurzolo: Making a mesh router/gateway from a smartphone: Is that a practical solution? Ad Hoc Networks (2011). Doi:10.1016/j.adhoc.2011.03.004, In Press
13. Kashyab, I., Rathy, R., Pandey, D.: A hierarchy based routing protocol for disaster recovery system. In: Optimization, Reliabilty, and Information Technology (ICROIT), 2014 International Conference on, pp. 217–221 (2014). DOI 10.1109/ICROIT.2014.6798317
14. Legendre, F., Theus, H., Felix, S., Bernhard, P.: 30 Years of Wireless Ad Hoc Networking Research: What about Humanitarian and Disaster Relief Solutions? What are we still missing? In: In proc. of the 1st ACM International Conference on Wireless Technologies for Humanitarian Relief (2011)
15. Martín-Campillo, A., Crowcroft, J., Yoneki, E., Mart, R.: Evaluating opportunistic networks in disaster scenarios. J. Network and Computer Applications pp. 870–880 (2013)
16. Milic, B., Malek, M.: NPART - node placement algorithm for realistic topologies in wireless multihop network simulation. In: Proceedings of the 2nd International Conference on Simulation Tools and Techniques, Simutools '09. ICST, Brussels, Belgium, Belgium (2009). URL `http://dx.doi.org/10.4108/ICST.SIMUTOOLS2009.5669`
17. Minh, Q.T., Nguyen, K., Kamioka, E., Yamada, S.: Tree-based disaster recovery multihop access network. In: Communications (APCC), 2013 19th Asia-Pacific Conference on, pp. 409–414 (2013). DOI 10.1109/APCC.2013.6765980
18. Moreno, Y., Pastor-Satorras, R., Vespignani, A.: Epidemic outbreaks in complex heterogeneous networks. The European Physical Journal B - Condensed Matter and Complex Systems **26**, 521–529 (2002). DOI 10.1140/epjb. URL `http://dx.doi.org/10.1140/epjb/e20020122`
19. Mourad, F., Chehade, H., Snoussi, H., Yalaoui, F., Amodeo, L., Richard, C.: Controlled mobility sensor networks for target tracking using ant colony optimization. Mobile Computing, IEEE Transactions on **11**(8), 1261–1273 (2012). DOI 10.1109/TMC.2011.154
20. Natalizio, E., Loscrí, V.: Controlled mobility in mobile sensor networks: advantages, issues and challenges. Telecommunication Systems pp. 1–8 (2011)
21. Natalizio, E., Surace, R., Loscrí, V., Guerriero, F., Melodia, T.: 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), pp. 319–323. Hangzhou, China (2013). DOI 10.1109/MASS.2013.40
22. Nelson, S.C., Harris III, A.F., Kravets, R.: Event-driven, role-based mobility in disaster recovery networks. In: Proceedings of the second ACM workshop on Challenged networks, pp. 27–34. ACM (2007)
23. Nishiyama, H., Ito, M., Kato, N.: Relay-by-smartphone: realizing multihop device-to-device communications. Communications Magazine, IEEE **52**(4), 56–65 (2014). DOI 10.1109/MCOM.2014.6807947
24. Ray, N.K., Turuk, A.K.: A framework for disaster management using wireless ad hoc networks. In: Proceedings of the 2011 International Conference on Communication, Computing & Security, ICCCS '11, pp. 138–141. ACM, New York, NY, USA (2011). DOI 10.1145/1947940.1947970. URL `http://doi.acm.org/10.1145/1947940.1947970`
25. Sakano, T., Fadlullah, Z., Ngo, T., Nishiyama, H., Nakazawa, M., Adachi, F., Kato, N., Takahara, A., Kumagai, T., Kasahara, H., Kurihara, S.: Disaster-resilient networking: a new vision based on movable and deployable resource units. Network, IEEE **27**(4), 40–46 (2013). DOI 10.1109/MNET.2013.6574664

26. Uddin, M.Y.S., Nicol, D.M., Abdelzaher, T.F., Kravets, R.H.: A post-disaster mobility model for delay tolerant networking. In: Winter Simulation Conference, WSC '09, pp. 2785–2796. Winter Simulation Conference (2009). URL http://dl.acm.org/citation.cfm?id=1995456.1995836
27. Younis, M., Senturk, I.F., Akkaya, K., Lee, S., Senel, F.: Topology management techniques for tolerating node failures in wireless sensor networks: A survey. Computer Networks (2013)
28. Zema, N.R., Natalizio, E., Poss, M., Ruggeri, G., Molinaro, A.: Healing wireless sensor networks from malicious epidemic diffusion. In: Distributed Computing in Sensor Systems (DCOSS), 2014 IEEE International Conference on, pp. 171–178. IEEE (2014)