Simulation of Emergency Response Operations for a Static Chemical Spill within a Building using an Opportunistic Resource Utilization Network

Leszek T. Lilien,1 Mohammed W. Elbes,1 Lotfi ben Othmane2 and Raed M. Salih1
1 Department of Computer Science Western Michigan University Kalamazoo, MI, USA
2 Department of Mathematics and Computer Science Eindhoven University of Technology Eindhoven, The Netherlands

Abstract—We investigate supporting emergency response operations with opportunistic resource utilization networks (“oppnets”), based on a network paradigm for inviting and integrating diverse devices and systems available in the environment. We simulate chemical spill on a single floor of a building and indicate how oppnets can assist first responders in finding optimal rescue routes for evacuating victims. The obtained results show how the number of victims rescued by the responders and the time required for their evacuation depend on: (a) the number of an oppnet’s helpers that the oppnet is able to integrate; (b) the number and distribution of victims within the incident area; and (c) the number of responders (their optimal number is identified). The results demonstrate that using oppnets can significantly improve efficiency and effectiveness of emergency response operations.

Keywords—emergency response operations; oppnets; opportunistic resource utilization networks; simulation.

I. INTRODUCTION

One of the important lessons from the large natural and man-made disasters—such as the tsunami in Japan and the Southeast Asia in March 2011, and the 9/11 terrorist attack—is the clear understanding of how much the efforts of emergency responders can be hampered by inadequate support from communication and computer facilities. Providing flexible and dependable communication and computing support is a critical requirement for saving more victims and for safety and efficiency of first responders.

We investigate supporting emergency response operations with a specialized computer network technology (and paradigm) proposed earlier by us; it is named opportunistic resource utilization networks or oppnets. An oppnet is capable of inviting and integrating diverse devices and systems available in the environment; the integrated devices/systems become the oppnet’s helpers. An oppnet can utilize all helpers’ resources for performing its tasks. By such integration, an oppnet grows from its original set of initial nodes employed together (the seed oppnet) into a larger network including numerous helpers (the extended oppnet).

The simulated scenarios consider a static (neither expanding nor shrinking) chemical spill on a single floor of a building. We assume that the number of victims is known (the security desk at the building entrance records all people entering and leaving).

The simulation scenario deals first with localization of all victims who are unable to self-evacuate, and then with their evacuation by rescuers. We assume a partial view of the building floor by sensors or cameras, which means that some victims remain unlocalizable by sensors or cameras. (This is a more general and more challenging case than the case of perfect view, which assumes a full coverage of the building floor by sensors or cameras; this means that exact locations of all victims are easily pinpointed.)

The cameras and sensors can be carried by the victims (e.g., in cellphones carried by them, in their pockets, embedded in their intelligent badges) or present in the rooms (e.g., surveillance cameras; presence detectors; in cellphones lying on desks; in desktops and laptops).

Evacuation planning requires finding optimal routes for evacuation of victims by rescuers. We indicate how oppnets can assist first responders in finding these routes. More specifically, the obtained results show how the number of victims rescued by the responders, as well as the time required for the evacuation depend on the number of an oppnet’s helpers that the oppnet is able to integrate, the number and distribution of victims within the incident area, and the number of responders (their optimal number is identified).

Based on these results, we conclude that oppnets have a direct impact on efficiency and effectiveness of emergency response operations: the duration of evacuation depends on the helpers’ communication via the oppnet.

The paper is organized as follows. Section II is a brief introduction to oppnets. Section III presents related work. Section IV describes the simulation scenario and setup. Section V provides results of the simulation experiments. Section VI concludes the paper.

II. A BRIEF INTRODUCTION TO OPPNETS

Opportunistic resource utilization networks or oppnets were invented by one of us for effective support of emergency response operations [1]. An oppnet behaves like an emergency team that, when in need of more resources to cope with an emergency, enlists helpers with additional resources.

The goal of oppnets is opportunistic leveraging of resources in possession of systems that are within oppnet’s reach. An oppnet starts its operation as a seed oppnet, a pre-designed yet ad hoc network (it is ad hoc in terms of node locations). Any system or device that an oppnet is able to contact and that possesses resources useful for the operation of the oppnet is potentially the oppnet’s helper (the useful resources include but are not limited to communication, computation, sensing, and storage resources). By integrating helpers into its activities, the seed oppnet grows into a much more capable expanded oppnet. We identify two categories of helpers: volunteers (who must be asked for help) and

978-1-4799-1535-4/13/$31.00 ©2013 IEEE 408
reservists (who are can be ordered to help). In this paper only the latter are assumed.

III. RELATED WORK

Many efforts of researchers and government agencies are aimed at designing and developing platforms and software for modeling and simulation of emergency preparedness and response. For example, as the result of these efforts, the Environmental Protection Agency (EPA) provides the software suite named Computer-Aided Management of Emergency Operations (CAMEO) [2]. CAMEO is a system of software applications developed by EPA’s Office of Emergency Management (OEM) and the National Oceanic and Atmospheric Administration (NOAA) Office of Response and Restoration to plan for and respond to chemical emergencies.

Maper et al. [3] developed a software environment named the Standard Unified Modeling, Mapping, and Integration Toolkit (SUMMIT) that integrates visualization, modeling, and simulation. SUMMIT allows users to access and visualize integrated data resources and modeling tools for planning, exercise, and operation response. It consists of three components: the SUMMIT web-based client, the SUMMIT server, and the SUMMIT-accessible data and models. SUMMIT has been deployed at regional and national level; it was used in the National Level Exercise 2012 (a cyber-attack scenario) in USA, and to support planning and response exercises (chemical release, flooding, and infrastructure attack scenarios) in Sweden.

Rossetti and Ni [4] use Quadstone Paramics, a microscopic simulation software for traffic and pedestrian transportation infrastructure, to present six simulations for people evacuation in a large shopping district around Northwest Arkansas Mall at Fayetteville, AR. The simulation focuses on the impact of evacuation of people (from parking lots to a safe area) on the traffic flows. The authors consider two factors: the occupancy rate of parking lots and the background traffic level, while they ignore other factors that may affect an evacuation (such as traffic operations, response rates, and evacuation sequences).

Filippoupolitis et al. [5] proposed a simulation to model evacuation of a multi-story building. Their simulation model uses a wireless sensor network to monitor the spread of hazards (fire) inside of a ten-story office building. Autonomous intelligent agents using Java Agent DEvelopment Framework (JADE) are used to represent actors and their interaction inside the building. A user can observe the simulation environment through a graphical user interface (GUI) that shows the main floor of the building. The GUI also allows a user to change (in real time) some simulation parameters—such as simulation speed (slow or fast), simulation time, and the number of actors. The simulation provides real-time decision support during the emergency situation. The authors show the effect of actors’ behaviors both as individuals and in groups, the effect of sensed information on resource allocation and decision making, and the impact of the leadership.

Delimghani and Rao [6] present a simulation measurement obtained from the use of a hybrid wireless ad hoc mesh network testbed [7]. The simulation is used to provide local communication for responders during a campus drill. The mesh network supports multiple interfaces that facilitate collaboration of different systems in a heterogeneous environment. The authors highlight technical challenges and requirements—such as the dependency of the communication on power supply, security, and privacy.

An earlier paper of our research team on oppnets [8], first defines oppnets, and then presents an implementation and using the Oppnet Virtual Machine (OVF) for natural disasters. To find survivors near the incident area, a seed oppnet is initially deployed. Then, the seed oppnet expands by integrating many diverse helpers (including serving policemen, firefighters, computing systems belonging to federal/local governments or local universities, etc.); all helpers contribute to improving performance of emergency response operations, resulting in saving more victims and better protecting first responders.

IV. THE SIMULATION SCENARIO AND SETUP

This section describes the simulation scenario, simulation modeling and assumptions, the parameters and variables for the simulation, and implementation issues.

A. The Basic Simulation Scenario

Suppose that a chemical spill or attack occurs in a building. Some people become victims: they are trapped in the threatened area. Some of them are incapacitated or unable to escape due to physical obstacles (e.g., trapped by damaged buildings or blocked door). Some are still adequately mobile but either disoriented or too scared to escape from the threatened area (for example, they are afraid of getting contaminated during their escape).

The rescue team arrives on the scene, and deploys a seed oppnet. The seed grows into an expanded oppnet by finding and enlisting helper systems and devices in the area. It can enlist devices in possession of victims, such as their laptops, cellphones, iPads, etc.

Once oppnet communications with a victim’s device is established, help messages from the victim (automatically including her position data from her GPS-equipped devices, whenever available) are forwarded to the oppnet’s Victim Location Database (ViLDa). ViLDa records are used by oppnet helpers running AVEEP (Automatic Victim Escape/Evacuation Planning) software to mark known locations of victims.

Next, AVEEP uses its maps for escape/evacuation planning, that is, for automatic identification of optimal access and evacuation routes. The AVEEP routes are used by rescuers that must carry out victims unable to self-evacuate. (The AVEEP routes are also used as escape routes for victims able to self-evacuate, who receive evacuation route maps and directions—also via the reverse 911 call out technology. But such victims are not considered in this paper.)

B. More Details for the Simulation Scenario

The scenario models rescuing victims who are unable to self-evacuate, and assumes a partial view of the building floor by sensors and cameras.

The simulated building. Figure 1 shows the simulated building. It includes seed nodes, cameras, sensors, victims able to self-evacuate and victims unable to self-evacuate.

The following acronyms are used in this figure: C – camera; Ei – i-th exit; Ri – i-th room; RR – rescuer; V – victim.
The following graphical symbols are used:

- **Small circles** (joined by lines) represent the vertices of the graph modeling the rooms of the building.
- **Lines** joining these circles indicate routes that can be used for evacuation.
- **Squares** at the ends of the lines, labeled with \( E1-E7 \) indicate all evacuation exits from the building.
- **Rectangles** (with electronic circuit images) represent sensors.

(Crosses with circles superimposed on them represent self-evacuating victims; such victims are ignored in this paper.)

**Simulations inputs.** The simulations inputs are:

1. Maps and plans of the affected building.
2. Identification of contaminated subareas (provided by sensors communicating with the expanded oppnet).
3. Locations of rescuers and their resources (can be continuously located and identified by means of, e.g., GPS, RFID, PDAs, cellphones, etc.).
4. Locations of non-communicating or unable to move victims, that is, the victims that are either unable to receive escape route information in any way or unable to move out of the endangered area to a safe area along the escape routes.
5. Specification of the coverage of the incident area by images and other sensing, which is needed to: (i) locate non-communicating or unable to move victims (i.e., all victims that require the assistance of a rescuer to be led or carried to a safe area); and (ii) to verify that planned rescue routes are passable.

**Simulation metrics.** We measure the following quantities:

1. The number of rescued victims (also as the function of the evacuation time, which is time to complete the evacuation of the last victims that the oppnet is able to locate, of necessity ignoring victims unlocalizable due to being out of camera/sensor ranges).
2. The percentage of victims that are not rescued due to being unlocalizable.
3. The relationship between the evacuation time and the density of cameras and sensors.

**C. Simulation Modeling**

We describe in the following our modeling of the entities locations and rescue paths, and assignments of responders to victims.

**Modeling entity locations and rescue paths.** In the graph model of the building, we use vertices and edges as follows:

- Vertices represent “area units,” such as rooms and staircases. Each vertex is described by a quadruple \((x, y, l, w)\), where \(x\) and \(y\) are the coordinates, and \(w\) and \(l\) are the width and length of the area unit.
- Edges represent paths between area units, e.g., a route between neighboring rooms or from a room to a staircase. Each edge is described by the triple \((s, d, l)\),

...
where \( s \) is the source area unit, \( d \) is the destination area unit, and \( l \) is the distance between \( s \) and \( d \).

**Modeling assignment of responders to victims.** We use Integer Linear Programming (ILP) [9] to model the assignments of responders to victims (unable to self-evacuate), and the LP solver QSopt [10] to find the optimal assignments.

Suppose that we have \( m \) rescuers and \( n \) victims. Each rescuer may be assigned to a set of victims. The ILP formulation of the optimal total distance for all rescuers to save all victims assigned to them is:

Minimize \[
\sum_{i} \sum_{j} d_{ij} x_{ij}
\]

\( x_{ij} = 1 \) iff Rescuer \( i \) is assigned to save Victim \( j \);

0 otherwise

Equation 2 is a constraint assuring that each victim is assisted by one rescuer. The input to the ILP formulation, the \( d_{ij} \) values, are calculated by the shortest path algorithm.

**D. Simulation Assumptions**

The simulation assumptions are:

A1. We have only a partial view coverage of the building floor by sensors and cameras.

A2. The chemical spill is static (neither shrinking nor expanding).

A3. The number of victims is known. (For example, the security desk at the building records all persons entering and leaving.)

Only locations of victims that are covered by cameras and sensors are known. Hence, due to Assumption A1, some victims remain unlocalized. They can not be helped by the rescuers.

Based on Assumption A2, the paths unavailable due to contamination are static (do not change over time).

**E. Parameters and Random Variables for the Simulation**

Table I lists and describes the random variables and their statistical distributions. Table II lists and describes the parameters used in the simulation.

**F. Implementation Issues**

**Simulation facilities.** The simulation was run on the Dell Inspiron laptop with Intel Core i3 processors, under MS Windows XP Pro (32 bit). The code in Java, using SimJava (an event-driven simulation package) [11] was executed in the NetBeans Integrated Development Environment.

We used two Java libraries in the simulation code: (i) JGraphT [12]—a free Java graph library, providing mathematical graph-theory objects and algorithms; and (ii) QSopt [10]—a linear programming solver. We utilized several algorithms from these libraries, including the Dijkstra shortest-path algorithm [13].

**Table I: Random Variables for the Simulation**

<table>
<thead>
<tr>
<th>Variable Role</th>
<th>Value Distribution</th>
<th>Justification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Victim’s location</td>
<td>Normal distribution</td>
<td>Non-uniformity of victims’ locations</td>
</tr>
<tr>
<td>Victim’s communication status (0 or 1)</td>
<td>Binomial distribution</td>
<td>Binary value and non-uniformity of communication status</td>
</tr>
<tr>
<td>Victim’s mobility status (0 or 1)</td>
<td>Binomial distribution</td>
<td>Binary value and non-uniformity of mobility status</td>
</tr>
<tr>
<td>Seed node location</td>
<td>Uniform distribution</td>
<td>Non-uniformity of seed node locations</td>
</tr>
<tr>
<td>Camera location (for non-fixed cameras)</td>
<td>Normal distribution</td>
<td>Non-uniformity of camera locations</td>
</tr>
<tr>
<td>Sensor location (for non-fixed sensors)</td>
<td>Normal distribution</td>
<td>Non-uniformity of sensor locations</td>
</tr>
<tr>
<td>Camera with a victim or not</td>
<td>Binomial distribution</td>
<td>Binary value and non-uniformity of camera being with a victim or not</td>
</tr>
<tr>
<td>Sensor with a victim or not</td>
<td>Binomial distribution</td>
<td>Binary value and non-uniformity of sensor being with a victim or not</td>
</tr>
</tbody>
</table>

**Table II: Parameters Values**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>( N ) – the number of simulation runs averaged for each result point</td>
<td>20</td>
</tr>
<tr>
<td>( n_{rescuers} ) – the number of rescuers</td>
<td>3 to 21 with step 3</td>
</tr>
<tr>
<td>( n_{victims} ) – the number of victims</td>
<td>5 to 20 with step 5</td>
</tr>
<tr>
<td>( n_{seeds} ) – the number of seed nodes</td>
<td>4</td>
</tr>
<tr>
<td>( n_{cameras} ) – the number of cameras</td>
<td>10 to 60 with step 10</td>
</tr>
<tr>
<td>( s_{range} ) – the camera range</td>
<td>2 to 6 with step 2</td>
</tr>
<tr>
<td>( n_{sensors} ) – the number of sensors</td>
<td>5 to 60 with step 5</td>
</tr>
<tr>
<td>( s_{range} ) – the sensing range</td>
<td>10</td>
</tr>
</tbody>
</table>

**Implementation challenges.** We modeled and implemented the properties and behavior of each seed node, camera helper, sensor helper, victim, and rescuer. Each of these objects receives input messages from other objects, acts upon them, and sends output messages to other objects.

The verification of the correctness of objects’ behavior was challenging due to the randomness of object properties and of initiation of messages.

To automate as much as possible so many simulation runs, we used the .bat file (a.bat) to generate all sequential “permutations” of the simulation variables. For each set of generated variables, we run the simulation 20 times, and then, to get a single data point for the figures, we averaged the results for these 20 runs.

The total simulation execution time for all seven experiments (shown in seven figures below) is almost three full days.

![Figure 2. The number of survivors vs. the number of sensors.](image-url)
V. EXPERIMENTAL RESULTS AND THEIR DISCUSSIONS

This section presents our simulation results. We use the parameters described in Table II.

In the experiments below, for a specific number of victims, we vary the number of rescuers, sensors, cameras, and helpers; and the camera range. The results explain how the number of survivors is affected by increasing each of these parameters.

Figure 2 shows that increasing the number of sensors generally increases the number of survivors (for a fixed number of cameras, rescuers and helpers; and the given camera range). This result is intuitive since a larger number of sensors means a higher probability of finding victims; with more victims located, the number of survivors grows.

Results of Figure 3 demonstrate the effect of the number of cameras on the numbers of rescued victims. It confirms in quantitative terms our expectations: the number of survivors (for a fixed number of sensors, rescuers and helpers) grows with the number of cameras.

In the experiment checking the effect of the camera range on the number of survivors, the camera range was varied from 2 m to 6 m for a fixed number of sensors, cameras, rescuers, and helpers (Figure 4). The effect of camera range is insignificant or small for all numbers of victims but one. An exception is the surprising case for the highest simulated number of victims (21 victims), in which increasing the camera range from 2 m to 6 m results in rescuing 20% more victims. We see no rational explanation for this case, and plan to run further experiments for more than 20 victims.

Availability of cameras and sensors does not mean that they will all be used as helpers in the evacuation process. The simulation results of Figure 5 present how the number of survivors is affected by utilizing a certain percentage of available helpers (cameras and sensors) for a given number of sensors, cameras, and rescuers. Consistently with expectations, we see that increasing the percentage of used helpers improves the survivor rate.

In the experiment checking the impact of the number of rescuers on the number of survivors (in Figure 6), we find that having more rescuers means more survivors. This is most clearly visible in cases when the initial ratio of rescuers to survivors is low, and then improves significantly (e.g., for 21 victims with initially just 5 rescuers only 8 victims are saved, while with finally 20 rescuers 12 victims are saved). Increasing the number of rescuers generally results in a higher survivor ratio, in part due to a higher reachability as rescuers are randomly distributed at the exits of the building.

Furthermore, as shown in Figure 7, when the total number of rescuers is lower than the total number of victims, increasing the number of rescuers reduces the total evacuation time (i.e., evacuation of all localized victims).

The final experiment investigates the number of rescuers who actually participated in the rescue operation out of the theoretically available number of rescuers waiting at the exits of the building (Figure 8). For example, when there are 21 victims in the building, only 7 rescuers out of 20 participate in the rescue operation, saving 12 victims. This means that some rescuers saved more than one victim while some other rescuers did not save any victim because they were blocked by obstacles (incl. building walls, the chemical spill, etc.) preventing their access to victims from their initial locations at the exits.
Due to their characteristics, use of opportunistic resource utilization networks or oppnets is most beneficial for an application or application classes characterized by the following properties: (i) the application can start with a small network, known as a seed oppnet; (ii) it requires high interoperability; (iii) it uses highly heterogeneous software and hardware components; (iv) it can benefit significantly from leveraging diverse resources obtained from helpers that are outside of the seed oppnet; (v) it is able to maintain persistent connectivity with helpers once it is established.

These characteristics are a good match for emergency response applications. Specifically, we have simulated use of an oppnet in rescue operations after a static chemical spill in a building. The results demonstrate that using oppnets can significantly improve efficiency and effectiveness of emergency response operations. Specifically, we have shown how use of different numbers of sensor and camera helpers, with various camera ranges, can impact rescue operation (incl. the number of victims saved, and the duration of the victim evacuation).

We plan to continue running more simulations of using oppnets for emergency response by: (i) extending the ranges of simulated parameter values; (ii) studying more control parameters; and (iii) extending the scenario to multi-story buildings (introducing the third dimension), and to campus or city district areas.

**ACKNOWLEDGMENTS**

We gratefully acknowledges contributions of Dr. Lilien’s students—Jamie L. Groos, Kruthika M. Kumar, and Padmini Nagaraj—to the earlier versions of the simulation program. L. Lilien thanks Dr. Mark Linderman and Mr. James Reilly, both from Air Force Research Lab, Rome, NY; the former—for discussions that triggered ideas incorporated in this report, including the important concept of using victim-provided image data in rescue operations; the latter—for discussions about sensor networks, sensor capabilities, and for emphasizing the potential of oppnets to minimize the need for infrastructure (which is replaced in oppnets by ad hoc helpers).

**REFERENCES**


