An Adaptive Emergency Protocol for People Evacuation in High-Rise Buildings

Soledad Escolar[a,1], David Villa[b], Félix J. Villanueva[b], Rubén Cantarero[b], Juan Carlos López[b]

[a] *Institute of Technology and Information Systems, Ciudad Real, Spain.*
[b] *Computer Science Department. University of Castilla-La Mancha, Ciudad Real, Spain*
[1]*Email: soledad.escolar@uclm.es*

***Abstract*—The occurrence of emergency situations in high-rise buildings, daily hosting hundreds of people, may force the massive evacuation of their occupants with the ultimate goal of preventing the loss of lives. In this paper, we propose an adaptive algorithm for dynamically computing safe evacuation routes, while the load of people is balanced between the accesses of each floor of the building, thus avoiding the accumulation of people in hotspots. The combination of this algorithm with sensors to detect the events of interest and a navigation system to guide the evacuees turns this solution into a specially suited approach for evacuation in high-rise buildings. Simulation results demonstrate that the yield evacuation routes drive all people outdoor with a similar average path length for the different risk scenarios addressed.**

***Keywords*-Smart Buildings; Intelligent Evacuation Systems; Wireless Sensor Networks.**

I. Introduction

Ambient intelligence, pervasive computing, and smart spaces are domains which have its main application area in buildings. Indoor environments have experienced a disruptive change in how we live and how we interact with the building, as result of innovation on ICT. One of the major issues still remaining, only partially supported, is the evacuation procedure on high-rise buildings under exceptional circumstances, e.g. fire, earthquake, or security threats. Smart buildings provide an excellent framework for implementing evacuation systems as they make an intensive use of sensors and actuators to improve comfortability, safety, and efficient consumption.

Intelligent evacuation systems [1] support the self-evacuation of civilians under the occurrence of emergency situations inside buildings, by dynamically determining the shortest routes towards the exit while preserving the safety of the evacuees. These systems integrate three main components [2]: 1) a monitoring system for anticipation/detection of emergency events such as fires, flooding, losses of pollutants, and threats; 2) an intelligent decision system that determines the safest evacuation route based on information obtained from the building infrastructure, its occupants, and from the detected-event propagation; and 3) a navigation system, generally based on lighting, to drive the evacuees towards the exit across the computed safe route.

State-of-art systems have traditionally overlooked that high-rise building tend to host thousands of people. This issue introduces a new difficulty when a single or few evacuation routes are provided without considering the amount of people evacuating through them. Addressing load balance in an evacuation route is a challenging topic, mainly due to the need to estimate the number of people and their location. Also, evacuation routes should consider the evolution of the risk to be avoided, so that escape routes should be updated with this information in seeking the safest path. In order to address this problematic, this paper proposes an intelligent dynamic evacuation system that considers safe and balanced evacuation routes as a major goal. After reviewing the Related Work we introduce the system model in Section III and the evacuation algorithm in high-rise buildings in Section IV. Section V presents the evaluation results and in Section VI we outline the conclusions and further research.

II. Related Work

A high-rise building is defined as a building that extends higher than the maximum reach of available fire-fighting equipment, which in absolute terms correspond to 23 meters above [3]. Today there exist, however, many samples of constructions that widely exceed such height. Let us just cite two examples: the 828 meters tall Burj Khalifa (Dubai) and the 632 meters of Sanghai Tower (China).

The evacuation of high-rise buildings promotes a 3D space searching problem that has typically been modeled using graphs. Computed graphs are afterward used to find the shortest path, still safe, between a location and an exit. To this end, the most common approach has been to use variations of Dijkstra's shortest path algorithm to deal with dynamic routes and with variable weights associated to each edge of the graph. The work in [4] presents a deep study of graph models and analytical models for emergency evacuation from a scalability and a simulation time perspective. With the evacuation time on mind, the authors use three algorithms to identify the most disruptive fire outbreak locations, determining the busiest node during an evacuation (to avoid congestion) and the total evacuation time (using analytical models). This theoretical work does not take into consideration different properties of the people and evacuation paths. Same authors propose, years later, in [5] two distributed and adaptive algorithms for driving people outdoor via less hazardous routes. The building is represented by means of a graph, where the vertices are locations where civilians can congregate (e.g. rooms, corridors) and the edges are physical paths that can be elected for transiting. Edges are labeled with a value that is a function of its length and an estimation of the risk. The proposed algorithms rely on static sensor nodes (SN) located at the edges for building monitoring; the first algorithm uses static decision nodes (DN) in each vertex while the second algorithm assumes mobile nodes (CN) that are carried by civilians (e.g mobile phones) forming an opportunistic network (*oppnet*). SN communicate their observations to DN or CN (in the first and second case, respectively) which compute the next hop with minimal cost within the path. Simulations demonstrate that the first algorithm outperforms better in terms of evacuation time and number of evacuated civilians. The suitability of using opportunistic communications for evacuation during the emergency has been evaluated in [6], whose experimental results conclude that *oppnets* may successfully support evacuation, especially in dense urban environments. In [7] a graph model is presented for estimating the Hazard, Shortest Path and Safety Path for fire emergency evacuation. In this work also the evolution of the fire is estimated and it is assumed that there is a sensor network that provide the control center with information about the hazard of specific paths. In [8] a simulation system devoted to support and to study the evacuation procedure is developed. Based on agent technology, the environment is modeled as a graph (each floor) connected through staircases. On exceptional situations, each edge of the graph represents the distance between two vertices (e.g rooms, doors, fire exits) and a hazard estimation of the path according to the current situation. This work assumes the presence of sensor nodes distributed along the building which provide with hazard estimation. A set of decision nodes (equipped with dynamic panels) provide users with the shortest path to exit minimizing the risk. Also information about the situation is transmitted to the mobile communication nodes of the users (e.g. smartphones). The building on this work is unaware of the number of users present at each space/room, also the model does not contemplate path particularities like going through a stairway. In the proposed solution there is no distinction between people capabilities (e.g elderly people or handicapped people).

The ability to combine human factors affecting the evacuation [9] with building infrastructure is the subject of [10], which focuses on an individualized evacuation under a fire occurrence in high-rise buildings. The authors simulate the behavior of an intelligent navigation system for evacuation in buildings that are modeled as 3D-GIS maps. Additionally, a set of 16 risk factors are identified along the three dimensions (human, fire, and infrastructure) that affect the response performance, as for example, age, gender, disabilities and physical conditions of evacuees, fire propagation and occupants density. Each factor takes a value of risk that is determined based on a priority and a value obtained as result of a survey done by 78 participants. The risk score is computed as the sum of the 16 risk factors and classified in one of five levels of risk. The system is then evaluated by means a Multilayer Perceptron network that predicts the risk level in the evacuation path, demonstrating an accuracy of 93.8%.

Differently to these works, this paper presents an evacuation for high-rise buildings that relies on two foundations: risk estimation to select the safer path towards the exit, since the shortest path is not always in line with the routes of minimal risk; and the knowledge of the number of people in each floor to balance the load avoiding overloaded critical points along the exit path.

III. System Model

Consider a utility building of $h+1$ floors ($h \geq 0$) that every day is accessed by many people including personnel, visitors, and public in general. The ground floor ($h = 0$) holds the main exit of the building. At each floor there exist several accesses through which individuals may reach that floor, as for instance the stairs, the emergency stairs, and the elevators. We denote with A_i the set of access points in floor $i \in [0, h]$ i.e. $A_i = \{a_{i1}, a_{i2}, \ldots, a_{ik}\}$, where k is the number of access points in floor i. For the sake of simplicity we will assume that k is equal for all floors excepting the ground floor, which holds an additional exit named a_{00}. Each floor is monitored by means of a Wireless Sensor Network (WSN) with a twofold aim: to detect emergency events (e.g. fire, explosion, gas leakage) and people detection. The WSN consists of networked sensor nodes distributed on each floor and of a gateway that ensures intra-floor communication. There exist two types of sensor nodes: event detectors and people tracers. The former are spread over the floor and are equipped with a gas sensor (smoke detector), a humidity sensor, and a temperature sensor. The latter are strategically located in the near surroundings of each access point and are equipped with a PIR sensor, a depth-finding camera (e.g.

Kinect) and an infrared barrier sensor, whose readings support the inference of the number of the people that are occupying the floor at each moment in time. We denote with P_i the number of individuals at floor i and with P to the total number of individuals in the building, computed as $P = \sum_{i=0}^{h} P_i$. Note that, according to the measures taken by the sensors, an access point a_{ij} knows its current occupation o_{ij}, which is the number of people that is actually using that access point (e.g. accessing or leaving the floor) and its maximum capacity C_{ij}, which is the maximum load supported by a_{ij} expressed in number of people (consider, for example, the maximum capacity of an elevator or the maximum number of people that can transit a stairway measured in people/m^2). Note also that $o_{ij} \leq C_{ij}$ must always hold.

An access point a_{ij} is directly connected to adjacent access points in floors i, $i-1$ and $i+1$. Let us to define a_{ij} and a_{yx} as adjacent iff $y = \{i-1, i+1\}$ and $x = j$ or $y = i$ and $x = \{j-1, j+1\}$ $\forall y \in (0, h)$, $x \in (1, k)$. The special point a_{00} is connected to all the remainder access points at floor $h = 0$. We model the adjacency relationship as a connected, acyclic graph $G = (V, E)$ with $V = \{A_0, A_1, \ldots, A_h\}$ and $E = \{(a_{ij}, a_{yx})\}$ with a_{ij} and a_{yx} adjacent. Therefore, G has $k(h+1)+1$ vertices and $(2k-1)(h+1)$ edges. Each edge $(a_{ij}, a_{yx}) \in E$ is labeled with a cost $w_{a_{ij}}^{a_{yx}}$, which expresses the cost to reach a_{yx} from a_{ij}. Then, given a graph G, which represents all possible paths between any access point and a_{00}, the evacuation problem is formulated as follows: for each floor i with $P_i > 0$ find the subgraph $G' = (V', E')$ to evacuate P_i individuals towards a_{00} with the minimum cost W, which is given by the sum of the costs of all its edges, i.e. $W = \sum w_{a_{ij}}^{a_{yx}} \forall (a_{ij}, a_{yx}) \in E'$. Section IV explains the cost function and introduces the employed heuristic for finding the (sub)optimal subgraph G' with minimal cost and balanced load.

Finally, to implement the physical evacuation of individuals, the building should be also provided with a navigation system based on light panels located in the near surroundings of each access point, with the ultimate purpose of guiding the evacuees towards the most adequate adjacent access point.

A. An Example of Building

The Escuela Superior de Informática (ESI) of University of Castilla-La Mancha is located in Fermín Caballero building, a construction of three floors high ($h = 3$). Each floor has five access points: two elevators, two stairs, and an emergency stair that connects all the floors. The ground floor ($h = 0$) has also an additional access point which is the main entrance/exit. A map of the first floor ($h = 1$) is depicted in Figure 1, where the access points are identified with red circles.

The second and third floor of the building present exactly the same architecture; analogously, they have the corresponding access points that match the points shown in the figure. Figure 2 a) shows the graph of connections of the building. Initial costs on the edges are statically computed, and, under the occurrence of an emergency event, are dynamically updated by considering sensor node observations. Figure 2 b) shows an example of an evacuation subgraph between a_{33} and a_{00} and cost $W = w_{a_{33}}^{a_{34}} + w_{a_{34}}^{a_{24}} + w_{a_{24}}^{a_{14}} + w_{a_{14}}^{a_{13}} + w_{a_{13}}^{a_{03}} + w_{a_{03}}^{a_{00}}$.

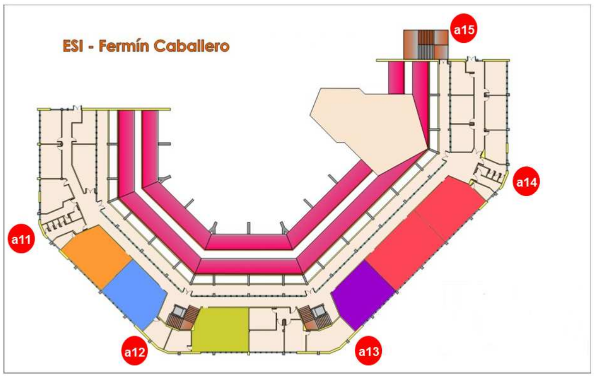

Figure 1. Map of 1st floor of ESI.

IV. Adaptive Evacuation System

This section presents the adaptive evacuation system proposed to dynamically determine, under an emergency situation, the evacuation plan for all the people in the building, through safe routes with minimal cost and balanced load. Our decision system relies both on the knowledge of the building infrastructure (static) and the observations that are periodically transmitted by the nodes regarding the risk and emergency event propagation (dynamic). This information is then used to update the cost between access points and take decisions on the most adequate direction for evacuation.

A. The Cost Function

Let us consider the graph of connection of a building $G = (V, E)$. Every edge $(a_{ij}, a_{yx}) \in E$ has a cost w that is computed taking into account the building infrastructure and the risk estimation. Specifically, the geographical distance d between a_{ij} and a_{yx} (if a_{ij} and a_{yx} are elevators we consider $d = 1$) and the effort e to be done by a person to achieve a_{yx} from a_{ij}: if a_{ij} and a_{yx} are in the same floor then $e = 1$; if a_{yx} is in the lower floor than a_{ij} then $e = 0.5$; otherwise $e = 1.5$. Let us define the risk level for an edge to take one out of the next five values: Low=1, Medium=2, High=3, Very High=4, and Critical=∞. Under normal conditions the risk level between any two access points is labeled as Low. However, under the occurrence of

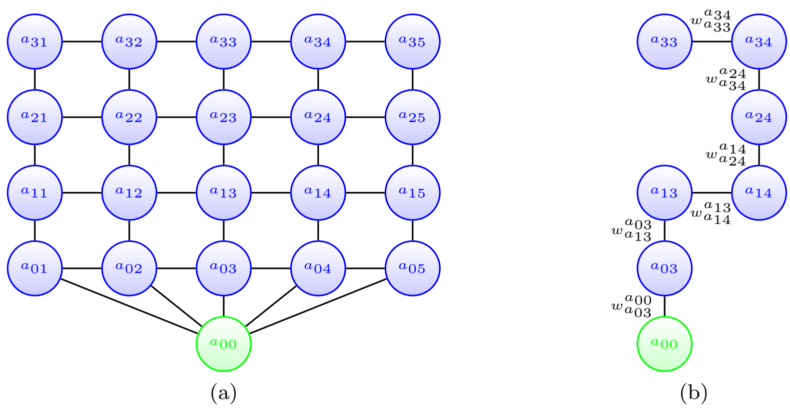

Figure 2. (a) Graph of Fermín Caballero Building; (b) Example of path (subgraph) between a_{33} and a_{00}.

abnormal events the risk level may increase based on the local observations done by nodes; when this happens, the decision system recomputes the new cost of the affected access points. A Critical risk is used to represent an edge that cannot be used for evacuation purposes. Note that a vertex a_{ij} with all its adjacent edges labeled as Critical induces a new subgraph $G' = (V', E')$ where $V' = V - a_{ij}$ and $E' = E - \{(a_{ij}, a_{(i+1)j}), (a_{ij}, a_{(i-1)j}), (a_{ij}, a_{i(j-1)}), (a_{ij}, a_{i(j+1)})\}$. Finally, the system considers the event propagation by defining the direction of propagation of the risk level r initiated at some access point a_{ij}. There exist four possible directions: up, bottom, right, and left. In a step of time, the emergency achieves the next access point a_{yx} found in the direction of propagation, leaving the edge between a_{ij} and a_{yx} with the risk level r.

Thus, the cost to achieve a_{yx} from a_{ij} comes given by $w_{a_{ij}}^{a_{yx}} = d \times r \times e$. Note that the cost function is an asymmetric function, i.e. $w_{a_{ij}}^{a_{yx}} \neq w_{a_{yx}}^{a_{ij}}$.

B. The Evacuation Plan

The pseudo-code to compute the evacuation plan is presented in Algorithm 1. After determining the access point in the highest floor $i \leq h$ with $P_i > 0$ and with the connection of lowest cost, the algorithm checks that the number of people in the building is not 0 and in this case initializes a subgraph G' and launches a recursive call to the function `eva`, which is presented in Algorithm 2. The function `eva` is a recursive function that computes the minimal cost route between the access point v_0 selected as input and the main exit a_{00}. When invoked, `eva` determines a new branch of the subgraph G' to evacuate the maximum number of people with the lowest cost. Upon achieving the ground floor, `eva` returns to Algorithm 1 and the process repeats until $P = 0$.

Function `eva` proceeds by initializing the set Δ with the adjacent vertices to the reference vertex v_0 and from Δ selects the one with minimal cost δ_{uv} in floor u (upper, lower, or equal floor than v_0) that becomes the next access point within the path towards a_{00}. If there not exists such a δ_{uv}, then the function `backtraking` is invoked to update the current subgraph by removing all vertices computed in the current route. Otherwise, the function continues by updating the counters of people in floors P_i and P_u. The graph G is updated by removing δ from V and (v_0, δ_{uv}) from E, the resulting graph G' is updated by adding δ to V' and the edge between δ and v_0 to E', and the cost of the solution is updated. The case base occurs when the ground floor is achieved $(i = 0)$, which means that P_0 people were evacuated, and the function comes back to Algorithm 1 by returning G' with cost W and the remaining people P'. Otherwise, a new call to `eva` will continue searching the next access point from δ_{uv} in the path towards a_{00}. Note that, when returning from `eva`, if P_i is still longer than 0, the next call to `eva` will select vertices different to those that compose the path previously computed (δ_{uv}) and edges with minimal cost but larger than $w_{v_0}^{\delta_{uv}}$, since δ_{uv} is

Algorithm 1 Evacuation Subgraph Algorithm

Require: G: connection graph.
Ensure: G': subgraph of evacuation towards a_{00};
Ensure: W: cost of G'.

```
Let P be the number of people in the building
Let V' = ∅; E' = ∅; W = 0
loop
  if P == 0 then
    return (G',W) % subgraph G' cost W
  Let i be the highest floor with P_i > 0
  Let Δ_ij = {(a_xy)} s.t. a_ij, a_xy are adjacent ∀a_ij ∈ A_i, j ≤ k
  Let v_0 = {a_xy} s.t min({a_ij^{a_xy}})∀a_xy ∈ Δ_ij, a_ij ∈ A_i, a_ij, a_xy are adjacent, and x = i;
  Δ_ij = Δ_ij − v_0
  V' = V' ∪ v_0;
  (G',W,P)=eva(G, G', W, i, v_0, P)
end loop
```

removed from the set of vertices of G, and $w_{v_0}^{\delta_{uv}} = \max$. This ensures that the evacuation process balances the load between access points of the same floor. The process repeats until P is evacuated and in each repetition a new path into the evacuation subgraph is generated.

Algorithm 2 eva Function

```
Require: G: connection graph.
Ensure: G': subgraph connecting an access point to a_00;
Ensure: W: the cost of G'.
Require: i: the floor
Require: v_0: the initial vertex.
Ensure: P: the number of people remaining.
  if i==0 then
    P = P − P_i
    return (G',W, P)
  Let Δ = {(a_xy)} s.t. v_0, a_xy ∈ V are adjacent
  Let δ_uv=min(w_v0^a_xy)∀a_xy ∈ Δ; u = x, v = y
  if ∄δ_uv then
    % The last path is not valid. Remove it from G'
    (G, G', W, i, v_0, P)=backtracking(G, G', W, i, v_0, P)
  else
    Let ⟨o_uv, C_uv⟩ be occupation, capacity of δ_uv
    % Update counters of people in floors i and u
    if i≠u then
      P_i = P_i − (C_uv − o_uv); P_u = P_u + (C_uv − o_uv);
    % Update G and G' and compute the cost W
    V = V − δ_uv; E = E − (v_0, δ_uv)
    V' = V' ∪ δ_uv; E' = E' ∪ (v_0, δ_uv)
    W = W + w_v0^δ_uv; v_0 = δ_uv; i = u; w_v0^δ_uv = max
    % Fulfill o_uv up to C_uv
    if C_uv − o_uv > 0 and P_i > 0 then
      o_uv = min(C_uv, P_i + o_uv)
  eva(G,G', W, i, v_0, P)
```

V. Evaluation

We evaluate our approach by simulation. To this end, we have developed a simulator that considers high buildings of $h \geq 8$ stories and any number of access points $k > 1$ at each floor. Each access point is randomly selected to be a lift or a stair, with a certain capacity and an initial occupation equal to 0. Two consecutive access points in a floor are located to a random distance d. For the sake of simplicity, the distances between two consecutive access points j and $j+1$ are equal for all floors and the capacity of the access point j at floor x is equal in the rest of floors, e.g. $C_{xj} = C_{ij}$ with $i \in [0, h]$. The building is occupied by a number of people P randomly distributed among its $h+1$ floors.

As explained in Section IV, all possible paths connecting any access point to the exit are represented by means of a graph. The cost to reach an access point (a vertex of the graph) from any of its four connected access points (adjacent vertices) is computed based on distance d, effort e, and risk level r between both. The objective of the simulator is to find the subgraph of minimal cost and load balanced that enables the evacuation of P persons under three scenarios: normal, static, and dynamic. The former represents the absence of an emergency event and thus the risk level measured between any pair of adjacent vertices is Low. The static scenario corresponds to an emergency event that is located in the proximities of an access point, which means that none of its adjacent vertices can be used for evacuation purposes since the risk level of its four edges is defined as Critical. The dynamic scenario simulates an emergency event that is progressively propagated in only one direction in the graph (up, bottom, left, or right). In each timeslot the detected event achieves the next access point found at the direction of propagation, which leaves invalid for evacuation all the edges of the achieved vertex (risk level is Critical). Figure 3 presents the resulting subgraphs for the three scenarios and for a building with $h = 8$ and $k = 5$, three of which ($j = 1, 2, 5$) are elevators with capacity equal to 12 persons and the other two ($j = 3, 4$) are stairs with capacity equal to 36, and $P = 400$ randomly distributed among the floors. The normal scenario results into a subgraph (on the left) where all minimal paths to achieve the exit are explored since the load is balanced across the access points. However, under the occurrence of an event located close to some access point, for instance at a_{55} (figure in the middle), the evacuation algorithm escapes such a vertex by avoiding to use any of its edges. In the dynamic scenario (on the right) the event is propagated from its initial position at a_{55} in up direction, thus achieving the vertices a_{65}, a_{75}, and a_{85}, each one in a single step of execution of the algorithm. Note, however, that a vertex may be selected while at the moment of the choice it is not affected by the emergency (a_{85} is achieved in step 4 but it could be used in any previous step). The algorithm greedily achieves the solution in several steps, each one providing a subgraph able to evacuate simultaneously the maximum number of people taking into account the capacity of the access points and the occupation of the floors. Table I details the subgraphs obtained in each step of execution of our algorithm in the three scenarios until evacuating P people. Columns 2, 3, and 4 show the parallel paths in a step of execution, the number of total vertices involved, and the number of people evacuated in each path, respectively. In each scenario, the paths computed can be represented as the subgraph sum (the subgraph of evacuation) that is depicted in Figure 3. At the end of the execution of the algorithm, the total number of people evacuated is equal to P.

A. Simulations

We have simulated our evacuation algorithm with high-rise buildings in the three scenarios proposed. To this purpose, we consider with $h \in [10, 100]$ and $k \in [6, 12]$, from which the ceiling number of $\frac{k}{2}$ are elevators

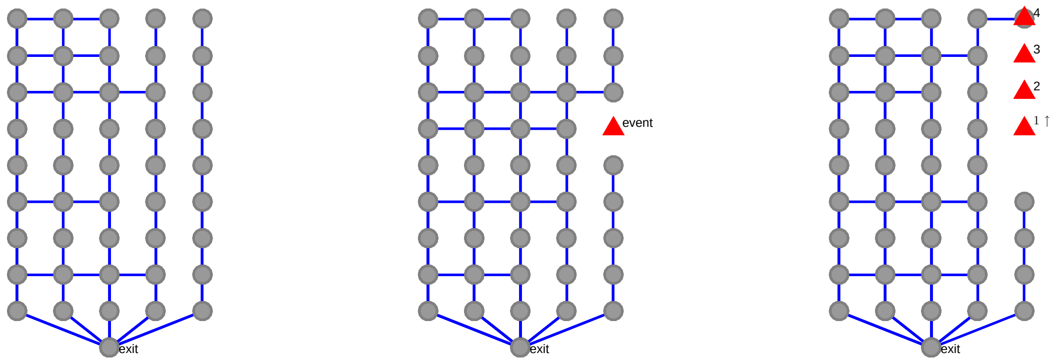

Figure 3. Evacuation subgraph in normal (left), static (center) and dynamic scenarios (right), $h = 8$, $k = 5$, $P = 400$.

Normal	**Subgraph**	**No. Hops**	**No. Evacuees**
Step 1	(8,0)…(0,0), (8,1)…(0,1), (8,4)…(0,4), (8,0)-(8,1)-(8,2)…(0,2), (8,3)…(0,3)	47	62,12,12,36,36
Step 2	(8,3)…(0,3), (8,0)…(0,0), (7,1)…(0,1), (7,4)…(0,4), (7,0)-(7,1)-(7,2)…(0,2)	44	36,12,12,12,36
Step 3	(6,0)-(6,1)-(6,2)-(6,3)…(0,3), (5,0)…(0,0), (4,1)…(0,1), (3,4)…(0,4), (3,0)-(3,1)-(3,2)…(0,2)	31	36,12,12,12,36
Step 4	(1,0)-(1,1)-(1,2)-(1,3)-(0,3)	5	26
	Total	**127**	**400**
Static	**Subgraph**	**No. Hops**	**No. Evacuees**
Step 1	(8,0)…(0,0), (8,1)…(0,1), (8,4)-(7,4)-(6,4)-(6,3)…(0,3), (8,0)-(8,1)-(8,2)…(0,2)	39	62, 12,36,36
Step 2	(8,3)-(7,3)-(6,3)-(6,2)-(6,1)…(0,1) (8,0)…(0,0)	20	12,12
Step 3	(8,1)…(0,1), (8,0)…(0,0)	18	12,12
Step 4	(7,1)…(0,1), (7,0)…(0,0)	16	12,12
Step 5	(7,1)…(0,1), (7,0)…(0,0), (6,0)-(6,1)-(6,2)…(0,2), (5,0)-(5,1)-(5,2)-(5,3)…(0,3), (4,4)…(0,4)	39	12,12,36,36,12
Step 6	(3,0)-(3,1)-(3,2)-(3,3)…(0,3), (1,0)…(0,0), (1,1)…(0,1), (1,4)…(0,4), (1,0)-(1,1)-(1,2)…(0,2)	17	36,12,12,12,2
	Total	**149**	**400**
Dynamic	**Subgraph**	**No. Hops**	**No. Evacuees**
Step 1	(8,0)…(0,0), (8,1)…(0,1), (8,4)-(8,3)…(0,3)	28	62,12,36
Step 2	(8,0)…(0,0), (8,1)…(0,1), (8,0)-(8,1)-(8,2)…(0,2)	29	12,12,36
Step 3	(8,2)…(0,2), (7,0)…(0,0), (7,1)…(0,1), (7,0)-(7,1)-(7,2)-(7,3)…(0,3)	36	36,12,12,36
Step 4	(6,0)-(6,1)-(6,2)…(0,2), (5,0)…(0,0), (4,1)…(0,1), (3,4)…(0,4), (3,0)-(3,1)-(3,2)-(3,3)…(0,3)	31	36,12,12,12,36
Step 5	(1,0)-(1,1)-(1,2)-(1,3)…(0,3)	5	26
	Total	**129**	**400**

Table I
PATHS OBTAINED IN EACH SUBGRAPH IN THE SCENARIOS NORMAL, STATIC, AND DYNAMIC.

with capacity for 12 people and the remainder number are stairs with a capacity for 36 people. In each experiment, we evaluate the evacuation of $P = 10000$ people randomly distributed among $h+1$ floors. The goal of the simulation is to compare the average number of access points or hops of the subgraphs of evacuation in the three scenarios. All the experiments were repeated three times, taking the average value to increase the accuracy of the results. However, we observe that for each experiment, the three results obtained could vary since the distribution of the people on the floors differs among repetitions of the same experiment, which impacts on the number of hops of the subgraph of evacuation.

Figure 4 shows the average number of hops in the normal (left), static (center) and dynamic scenario (right). As observed, the factor that mainly impacts on the number of hops is the number of floors in the building (x axis), since we have to compute the subgraph of evacuation from the highest occupied floor. However, the increase in the number of access points at each floor does not always result in an increase of the number of hops. Note that there is not observed a meaningful increase of the average number of hops between the different scenarios considered, since the algorithm is still able to find the subgraph of evacuation keeping the number of hops while the total cost of the subgraph is slightly increased. This fact is shown in Figure 5, where we show the average cost of a subgraph of evacuation with $h = 50$ for the three scenarios considered. As observed, the highest cost is obtained by the dynamic scenario, since such a subgraph use routes with highest risk and, therefore, with a larger cost. The same trend is repeated

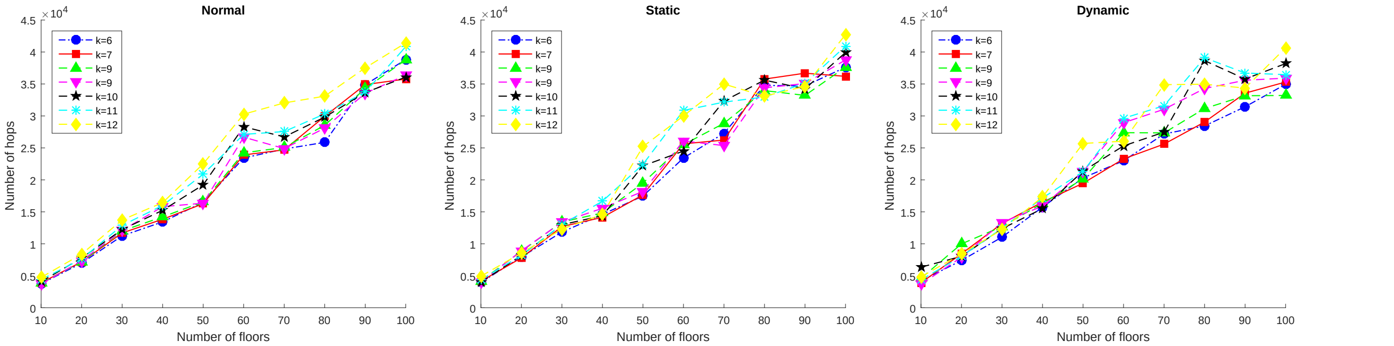

Figure 4. Average number of hops in the normal (left), static (center) and dynamic scenario (right).

in all the experiments.

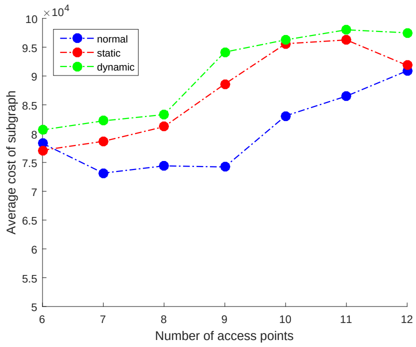

Figure 5. Average cost of the subgraph with $h = 50$

B. *Experimentation*

An actual deployment of the WSN presented in Section III would imply an important amount and variety of sensor devices for presence detection (e.g. passive infrared, barrier, sound) and it will probably require some actuators to allow the system to warn people and modify their behavior. In this way, a better path may be prioritized to achieve a model matching result. This kind of deployment is currently and on-going work in our research team.

The proposed model implies a key sensing problem that must be tackled: people counting, which may be a non-trivial problem involving computer vision, spurious data management, or high degree of uncertainty. Fortunately, we can focus just on the access points at each floor, where lighting conditions and people behavior are quite predictable. As a first approach, we use depth-finding cameras to segment people on the scene and get a reliable counter. For this purpose we use Microsoft Kinect 2, which provides a software library capable of assigning unique identifiers to bodies while tracking them in real time. From the obtained 3D coordinates, it is relatively easy to determine when a person is entering or leaving the access point defining two thresholds. They are the highlighted yellow squares in the Figure 6.

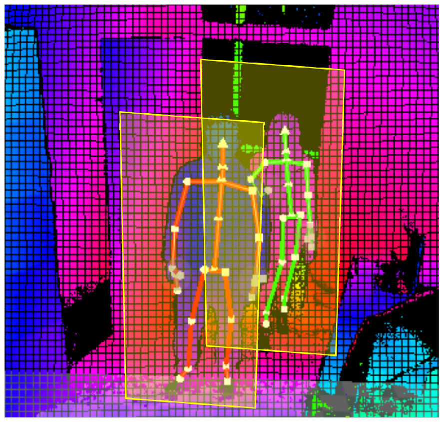

Figure 6. Monitoring an access point using Kinect

VI. Conclusions

This paper presents a strategy to evacuate the civilians inside high-rise buildings from their current locations towards the exit under the occurrence of an emergency event such as a fire, a flooding, or a security threat, initiated at any floor of the building. To this end, we have modeled high-rise buildings as a graph, whose vertices correspond to the places destined to enter/leave a floor (access points) and whose edges are connections between these places. Each edge is assigned a weight that represents the cost to achieve a destination from the source. The algorithm proposed here computes the evacuation subgraph as a set of the routes that evacuate all the people outdoor, with the minimum cost, e.g. selecting safe routes while people are balanced among different access points to avoid accumulation of people that can lead to risk situations. Simulation results demonstrate that the average number of hops of the subgraph of evacuation increases with the number of floors, while it is kept across different scenarios with the same configuration of k and h. As future work we plan to enrich the cost function with parameters related to human factors such as the age, disabilities and constraints of people that could impact on the effort to achieve an access point. A real deployment of the WSN able to implement the evacuation system proposed is also being installed at Institute of Technology and Information Systems, Ciudad Real (Spain).

Acknowledgments

This work has been funded by the Programme for Research and Innovation of University of Castilla-La Mancha, co-financed by the European Social Fund (Resolution of 25 August 2014) and by the Spanish Ministry of Economy and Competitiveness under project REBECCA (TEC2014-58036-C4-1-R) and the Regional Government of Castilla-La Mancha under project SAND (PEII_2014_046_P).

References

[1] K.-M. Yu, C.-S. Yu, C.-C. Lien, S.-T. Cheng, M.-Y. Lei, H.-P. Hsu, and N. Tsai, "Intelligent evacuation system integrated with image recognition technology," in *Ubi-Media Computing (UMEDIA), 2015 8th International Conference on*, Aug 2015, pp. 23–28.

[2] Q. Zhang, T. Chen, and X. zhi Lv, "New framework of intelligent evacuation system of buildings," *Procedia Engineering*, vol. 71, pp. 397 – 402, 2014, 2013 International Conference on Performance-based Fire and Fire Protection Engineering, Wuhan (ICPFFPE 2013).

[3] NFPA's High-Rise Building Safety Advisory Committee, "Guidelines to developing emergency action plans for all-hazard emergencies in high-rise office buildings," http://www.nfpa.org/~/media/files/safety-information/for-consumers/occupancies/highrise/emergencyactionplanhighrise.pdf?la=en, 2013.

[4] A. Desmet and E. Gelenbe, "Graph and analytical models for emergency evacuation," in *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2013 IEEE International Conference on*, March 2013, pp. 523–527.

[5] G. Gorbil, A. Filippoupolitis, and E. Gelenbe, "Intelligent navigation systems for building evacuation," in *Computer and Information Sciences II*, E. Gelenbe, R. Lent, and G. Sakellari, Eds. Springer London, 2012, pp. 339–345.

[6] G. Gorbil and E. Gelenbe, "Opportunistic communications for emergency support systems," *Procedia Computer Science*, vol. 5, pp. 39 – 47, 2011, the 2nd International Conference on Ambient Systems, Networks and Technologies (ANT-2011) / The 8th International Conference on Mobile Web Information Systems (MobiWIS 2011).

[7] T. Tabirca, K. Brown, and C. Sreenan, "A dynamic model for fire emergency evacuation based on wireless sensor networks," in *Parallel and Distributed Computing, 2009. ISPDC '09. Eighth International Symposium on*, June 2009, pp. 29–36.

[8] A. Filippoupolitis, G. Gorbil, and E. Gelenbe, "Pervasive emergency support systems for building evacuation," in *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2012 IEEE International Conference on*, March 2012, pp. 525–527.

[9] N. Pelechano and A. Malkawi, "Evacuation simulation models: Challenges in modeling high rise building evacuation with cellular automata approaches," *Automation in Construction*, vol. 17, no. 4, pp. 377 – 385, 2008.

[10] U. Atila, I. R. Karas, M. K. Turan, and A. A. Rahman, "Design of an Intelligent Individual Evacuation Model for High Rise Building Fires Based on Neural Network Within the Scope of 3d GIS," *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, no. 1, pp. 13–24, Sep. 2013.