2011 Fifth IEEE Conference on Self-Adaptive and Self-Organizing Systems Workshops

Spatial Computers for Emergency Management

Avgoustinos Filippoupolitis, Gokce Gorbil and Erol Gelenbe
Department of Electrical & Electronic Engineering
Imperial College London
London, UK
Email: {afil, g.gorbil, e.gelenbe}@imperial.ac.uk

***Abstract*—The evacuation of a building during an emergency situation, such as a fire, is a complex and challenging task. As the conditions inside the building change due to the spreading of the hazard, it becomes difficult for an evacuee to find the best evacuation path. Information systems can prove beneficial for the evacuees, as they provide them with directions regarding the best path to follow at any given time. In this paper we present two spatially distributed computing systems that operate inside a building. They adapt to the dynamic conditions during an evacuation while relying on local communication and computation for determining the best evacuation paths. The first system is composed of a network of decision nodes (DNs) positioned at specific locations inside the building. Their goal is to provide the evacuees with directions regarding the best available exit. The second system is composed of mobile communication nodes (CNs) carried by the evacuees. They form an opportunistic network in order to exchange information regarding the hazard and to direct the evacuees towards the safest exit. Sensor nodes that monitor the hazard intensity in the building are used by both systems. We use a multi-agent simulation platform that we developed to evaluate the performance of our proposed systems in evacuation scenarios inside multi-storey buildings. We show how parameters such as the frequency of information exchange between the nodes and communication ranges affect the performance of the systems.**

***Keywords*-Adaptive systems; emergency simulation; disaster management; building evacuation; opportunistic communications.**

I. Introduction

The evacuation that takes place inside a confined space, such as a building, is a complex situation. The occupants have to quickly decide which path to follow in order to exit the building safely. This, however, is not an easy task, especially if there is an ongoing hazard present. In this case, conditions can change rapidly as the state of paths may deteriorate with time. One of the main problems that evacuees face during an emergency situation is the lack of knowledge regarding the conditions in other parts of the building [1]. Most of the times there is ambiguous information with respect to which evacuation paths are safe and which are the locations that are affected by the hazard. This can lead to a delay in commencing the evacuation of the building and can also result in choosing an inappropriate evacuation route.

There are various approaches regarding the problem of movement decision support during emergency situations. In [2] the authors propose a distributed algorithm for robot navigation using a sensor network. They evaluate their approach using a robot and a sensor network composed of nine nodes. They do not, however, take into account other parameters such as a dynamically spreading hazard or high number of evacuees, which play an important role in providing decision support during a disaster. The authors in [3] propose an algorithm inspired by sensor network routing, in order to guide a flying robot. Although they also evaluated their method for guiding humans, the evaluation scenario included only one human and twelve sensors positioned inside a building. Scenarios with larger building occupancies and dynamically changing conditions were not investigated. In [4] a system based on sensor networks is proposed, for navigating the user to a goal location by avoiding hazardous areas. The path calculation algorithm is based on artificial potential fields. A testbed of 50 wireless nodes was used to evaluate the approach. The focus of the system was on the time needed by the nodes to obtain the shortest path. The authors, however, did not include an evaluation scenario with a spreading hazard or a large number of evacuees. A distributed navigation algorithm geared towards emergency situations is presented in [5]. The approach is inspired by an ad-hoc network routing protocol and uses hop-count as the distance metric. The authors tested their method using simulations in various network topologies and hazard locations. Although the algorithm was able to find exit paths avoiding hazardous areas, the spreading of the hazard and the presence of evacuees inside the area were not taken into account.

The aforementioned approaches mostly focus on how efficiently a (sensor) network can find a path towards a location of interest. Although the notion of hazard is present in most of them, the context in which each system functions is not directly related to an emergency situation. A common assumption is a static hazard representation and the absence of evacuees that interact with the system. We therefore propose two spatially distributed computing systems to provide intelligent navigation services to people for the purposes of safe and quick evacuation in dynamic hazards. The proposed systems are composed of small, self-contained and self-powered computing devices spatially distributed in a building. These devices, which we generally call *nodes*,

978-0-7695-4545-5/11 $26.00 © 2011 IEEE
DOI 10.1109/SASOW.2011.13

are capable of short-range wireless communications and they monitor the hazard, disseminate information on the hazard and the environment as it changes, calculate the "best" direction towards an exit and communicate their directions to the evacuees. By observing the current situation and updating their view of the environment through local communications, these systems are able to self-adapt as the hazard spreads, and continue to provide updated information and guidance to the civilians.

II. Description of the Spatial Computing Systems

We propose two different spatial computing systems that are able to provide dynamic evacuation directions to people in a building in case of an emergency such as a fire. Being spatial "computers", the design of our adaptive systems is closely related to the spatial characteristics of the operational environment. We therefore first state our assumptions regarding the physical environment and then describe our proposed systems.

A. Design Assumptions

We represent the building as a graph $G(V, E)$, where vertices V are locations where civilians can congregate, such as rooms, corridors and doorways, and edges E are physical paths that civilians will travel along while moving inside the building. The length $l(i,j)$ of an edge is the physical distance between vertices $i, j \in V$ while $H(i,j)$ represents the hazard intensity along this edge. We also define the "effective" length $L(i,j)$ of an edge: $L(i,j) = l(i,j) \cdot H(i,j)$. This metric expresses how hazardous an edge is for a civilian that will traverse it. When there is no hazard along the edge, $L \equiv l$ and the effective length becomes equivalent to the physical length of the edge. As the value of H increases, the corresponding edge becomes more hazardous to traverse.

We assume that the building graph is known for a building. We also assume that there are **sensor nodes (SNs)** installed in the building, where each SN monitors a graph edge as depicted in Fig. 1. Each SN has a unique device ID, a location tag that corresponds to the area (i.e. edge) it monitors, and short-range wireless communication capability so it can relay its measurements to other entities in the system. We assume that SNs are simple devices with low computing power and memory capacity. They do not perform any data storage or decision making. Each measurement is stored until it is over-written by a newer measurement. When a DN or CN requests the current measurement from an SN, the SN sends it its $H(i,j)$ value.

B. Intelligent Evacuation System (IES)

Our proposed intelligent evacuation system (IES) consists of static **decision nodes (DNs)**, which are installed at specific locations inside the building. In this discussion, we assume that a DN is positioned at each graph vertex. In practice, however, there could be fewer DNs, with each DN being in charge of providing directions for a contiguous set of locations. Each DN has short-range wireless communication capability, some local processor and memory, and a dynamic visual panel to present directions to civilians. If each civilian is equipped with a wireless hand-held device (such as a PDA), then DNs can also present their directions via wireless communications with such devices.

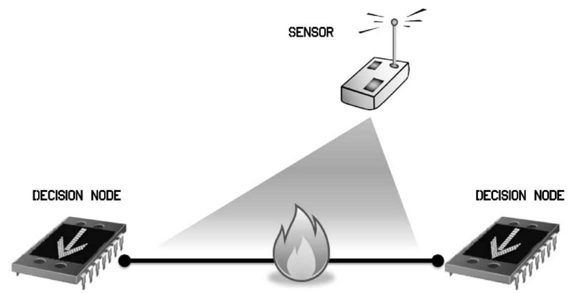


Figure 1. A sensor node (SN) monitors a graph edge for possible hazards. In the IES, decision nodes (DNs) located at graph vertices receive measurements from their adjacent SNs and provide dynamic directions during evacuation.

The goal of the IES is closely related to the system's spatial structure, since each DN computes the best direction towards the DNs located at building exits and communicates this (visually or via wireless) to the evacuees in its vicinity. DNs form a wireless network among themselves to exchange information in a distributed manner so they can provide up-to-date dynamic evacuation directions to civilians as the hazard spreads. Hazard information is provided to DNs by their adjacent SNs, and this information is further propagated among DNs based on the distributed decision algorithm as presented below. The definition of the effective length $L(i,j)$ only takes into account the hazard present along (i,j) as monitored by the associated SN [6]. We can, however, take advantage of the spatial characteristics of hazard spreading inside a confined space in order to improve the performance of our system. Our approach is inspired by the fact that a hazard, such as fire or smoke, that is present in a building location will affect neighbouring locations as time progresses [7], [8]. When DNs use hazard information as obtained only from their neighboring SNs (i.e. their adjacent links), hazard values from nearby SNs (non-adjacent but nearby links) are not taken into account during path calculation, which result in the calculation of paths that may direct evacuees towards locations where there is a nearby hazard. As the hazard spreads, it will eventually affect these locations and the evacuation path that passes through them. When this happens, the IES will calculate a new evacuation path and evacuees will be given new directions based on the new path. Waiting for the hazard to spread to calculate a new evacuation path can cause an increase in evacuation time since the evacuees will have to head to a new destination. Moreover, their health will be affected as they may come in contact with hazardous path sections during the re-calculation. By incorporating spatial hazard information in the effective edge length, the algorithm can proactively exclude paths that involve travelling near hazardous areas. To achieve this, we let each SN communicate with its neighbours and incorporate their readings into the "spatial" hazard value H_{sp} that is reported. The number of neighbours with which a sensor can communicate is affected by the SN communication range R.

Let us now give a new definition for the adjusted effective length of a link (i, j), $L_{sp}(i, j)$, which will include the spatial hazard information. Let m be a sensor measuring the hazard level H_m on link (i, j). A sensor n measuring the hazard level H_n on a link (i', j'), will then belong to the neighbours set $N(m)$ of m when $d(m, n) \leq R$, where $d(m, n)$ is the Euclidean distance between n and m and R is the communication range of the SNs. The effective length $L_{sp}(i, j)$ that includes the spatial hazard information is given by $L_{sp}(i, j) = l(i, j) \cdot H_{sp}(i, j)$, where

$$H_{sp}(i, j) = H(i, j) + \frac{1}{|N(m)|} \sum_{k \in N(m)} H_k \ .$$

Each DN, positioned at vertex u, stores the following information:

- the effective edge lengths to neighbors: $L_{sp}(u, n), \forall n \in V \mid (u, n) \in E$
- the effective lengths of the paths to an exit for all neighbors: $L_{sp}(n, e)$, $\forall n \in V \mid (u, n) \in E$ and e is a building exit,
- the effective length of the shortest path (SP) from u to an exit e: $L_{sp}(u, e)$,
- the next suggested DN d (i.e. the next hop along the SP from u to an exit).

It is not necessary for a DN to keep information regarding the effective length of the paths to all available exits. As the algorithm is executed, this information is propagated from the exits to the DNs and each DN will eventually select the best exit. The distributed decision algorithm, given in Alg. 1, is executed periodically by each DN. The algorithm is based on principles developed in [9], [10], and inspired by the distributed shortest path algorithm [11] and adaptive routing techniques such as the Cognitive Packet Network [12]. Its output is the next hop (i.e. DN) towards the nearest building exit. As edge costs are a combination of physical distance and hazard intensity, the paths calculated by the IES minimize travel distance to the exits while avoiding dangerous areas in the building.

The initial values for the effective length at each DN depend on its location inside the building. Exit DNs set their $L(u, e)$ values to 0, while all other DNs set it to ∞. The output of the algorithm is a suggestion towards the evacuees that are located near a DN. The suggestion is of the form **"go to v"**, where v is one of the neighbour DNs of u. Note that

Algorithm 1 Distributed calculation for the effective length $L_{sp}(u, e)$

Send to every neighbour n of u, the effective length of the path from u to the exit e : $L_{sp}(u, e)$
for each sensor node monitoring a link incident to u **do**
  Request hazard intensity H_{sp} from sensor node
  Calculate the effective length $L_{sp}(u, n)$,where n is a neighbour of u
end for
Update the effective length $L_{sp}(u, e)$ of the shortest path to the exit:
$L_{sp}(u, e)$ = min $\{L_{sp}(u, n)$+$L_{sp}(n, e)$: $\forall$ neighbours n of $u\}$
Set the next suggested Decision Node v:
v = argmin $\{L_{sp}(u, n)$ + $L_{sp}(n, e)$: $\forall$ neighbours n of $u\}$

the IES forms a spatial computing system where all sensing, computing and communication components of the system are static. The spatial characteristics of the building affect where these components are located. System parameters, such as the SN range R and the execution frequency of Alg. 1 by each DN, affect the performance of the system as we discuss in Sec. III-A.

C. *Opportunistic Communications based Evacuation System (OCES)*

The opportunistic communications based evacuation system (OCES) is composed of **mobile communication nodes (CNs)** carried by civilians. In the OCES, we assume that each civilian is equipped with a pocket- or hand-held device, with storage and processing capacity that would be equivalent to a mobile phone or similar unit, capable of short range (up to 10m) wireless communication. CNs form a network in an opportunistic manner as devices come into contact as a result of the vicinity of other humans and their mobility. Opportunistic communications (oppcomms) are characterized by the "store-carry-forward" paradigm [13] where CNs carry messages in local storage and then forward it to others when they get in communication range. Thus, a message is delivered to its destination via successive opportunistic contacts. Because the opportunistic network (oppnet) can be disconnected for long periods of time, CNs may need to carry messages for long durations and delivery of messages is not guaranteed.

Oppcomms are used to disseminate **emergency messages (EMs)** containing information on the hazard (i.e. location and intensity) among CNs. A CN obtains hazard observations from SNs in its vicinity, which are then translated by the CN into EMs that include the CN ID, locations (e.g. edges), intensities and timestamps of the hazard observation(s). An EM is disseminated among all CNs in the OCES, meaning each EM is sent network-wide. The first hazard observation or EM received by a CN acts as an alarm, indicating that there is a hazard and the civilian should evacuate the building. Each received EM is used to update the edge costs stored locally by a receiving CN, and triggers re-calculation of its local evacuation SP. The evacuation SP from the current CN location to the nearest building exit is calculated using Dijkstra's SP algorithm. Since effective edge lengths ($L(i, j)$ values) are used in SP calculation, the "shortest" path minimizes exposure to the hazard while also minimizing travel distance.

A CN uses its evacuation SP to provide a navigation service to its civilian by guiding her towards the next hop (i.e. graph vertex) on the SP. CNs use the SNs to find their location in the building. CNs request the location tag from their nearby SNs as they move within the building, and each near-by SN sends back a **localization message (LM)** which contains its location (or the monitored area, i.e. edge). CNs can then find out where they are in the building based on these LMs. The actual position of a CN is therefore approximated by its inferred location (vertex) on the building graph. **Epidemic routing (ER)** [14] is used for the dissemination of EMs in the oppnet. We have found that although ER is a classic routing algorithm for oppnets, it is very suitable for the OCES due to its flooding-based approach which closely matches how EMs should be disseminated, and its high message delivery ratio and low message latencies [15], which are critical in emergency communications. In order to store EMs, CNs employ *timestamp-priority queues*, where EMs with the earliest creation timestamps are dropped from the queue when it is full.

The OCES is a spatial computing system similar to IES, with the significant difference that the main computing and communication components of the system are mobile. This mobility introduces another level of dynamism to the system as communications are now affected by human mobility (which is of course restricted by the physical environment) and human mobility is in turn affected by the oppcomms since CNs depend on oppcomms to exchange and disseminate information on the hazard, which affects calculation of evacuation paths. We evaluate the effect of one environmental and one system parameter, population density and CN range respectively, on the performance of OCES in the next section.

III. Evaluation of the Proposed Spatial Computing Systems

We have implemented our proposed spatial computing systems inside the Distributed Building Evacuation Simulator (DBES) [16], [17]. The DBES is an agent-based discrete-event simulation platform for the simulation of emergency scenarios in confined and outdoor urban areas. Each actor (e.g. civilian, computing device, emergency response personnel, etc.) is represented as an agent with its own behaviour models (e.g. health and movement models for civilians). The physical environment (e.g. the building) is represented as a graph in the DBES as described in Sec. II-A. For each simulation run, the initial locations of people are chosen from a uniform distribution over the respective graph vertices within each floor to help evaluation of system performance in different occupancy patterns. We also change the spreading pattern and rate of the hazard (i.e. fire) in each simulation run, which is based on the probabilistic hazard model in the simulations [7], [16]. We do not simulate psychological aspects of people during an emergency and assume that people act rationally and follow directions given to them by the system (IES or OCES). When there is no evacuation system in use, we assume that the evacuees are familiar with the building (i.e. they know the whole building graph) and are able to calculate and follow the shortest path that leads to an exit. In this scenario, all civilians start to evacuate as soon as the fire starts. If an evacuee encounters fire (or smoke) during evacuation, she updates her knowledge of the building (i.e. the graph edge cost(s)) and re-calculates her SP. While the assumptions for the no-system scenario may be unrealistic in that they require too much from the people during an emergency, it provides a valuable benchmark with evacuation performance which is at least as good as what would normally be observed in a real-life scenario without any system.

A. Simulation Results: IES

We have evaluated the IES in an emergency scenario taking place inside the three-storey building. A fire erupts on the ground floor of the building and the occupants evacuate as soon as the fire starts using the four exits located on the ground floor. The population density is 20 people per floor. Each data point is the average of 100 simulation runs. Simulation results regarding the performance of IES with different SN ranges and algorithm execution frequencies are shown in Fig. 2. We use the ratio of fatally injured civilians to all building occupants and the average health of evacuees as the performance metrics.

We first comment on the effect of the sensor range (R) on the evacuation procedure. We used three different ranges, $R = \{1, 2, 3\}$ m. The results obtained for the smallest range (1m) are worse than the case where $R =$ 2m but better than the case $R =$ 3m. This indicates that there is a threshold value for R, until which increasing R increases system performance due to the inclusion of larger spatial areas in the effective link cost L_{sp}. After this threshold, system performance deteriorates with increasing R. The reason for this behaviour is the IES gets more conservative with increasing R in its evacuation path calculations. After a certain R value, which in this case lies between 2m and 3m, the conservative path calculation leads to safe paths being discarded as potentially dangerous by the IES. By discarding paths that are safe at an early stage during evacuation, the IES causes a higher number of people to become trapped in the building. The execution frequency of the distributed algorithm by the DNs also affects system performance, since the propagation of the changes in the environment depends on it. Although a high value for the algorithm execution frequency increases communication and computation costs, it also results in a more adaptive system which is able to give fast, correct suggestions to the evacuees. Simulation results indicate that the proper selection of the system parameters R and execution frequency may be non-trivial and depends on the spatial characteristics of the environment.

B. Simulation Results: OCES

We have evaluated the OCES in an emergency scenario taking place inside a three-storey building. A fire erupts on the 2nd floor of the building and the occupants evacuate using the two exits located on the ground floor. We assume that there is no fire alarm in the building (e.g. it has failed due to power outage) and people use the OCES as a fire alarm and a navigation system as described in Sec. II-C. SN range is set to 2m in the simulations and spatial information is not used. Each data point is the average of 50 simulation runs and results are shown with their 95% confidence intervals. Figure 3 shows system performance with different population densities (10–40 people per floor) and CN ranges (4–10m). We use average evacuation time (the average of the evacuation time for successfully evacuated civilians) and the ratio of successfully evacuated people to all building occupants as performance metrics.

We observe that increasing the maximum communication range of the CNs increases system performance. As can be expected, increasing CN range increases connectivity of the CNs, which results in better dissemination of hazard information among the CNs. Population density has a more complex effect on performance: increasing population density increases average evacuation time in all CN ranges. This can be expected since evacuation will take longer when there are more people in the building due to the higher level of congestion during evacuation (e.g. at the staircases). However, increasing population density does not always lead to higher evacuation ratios, especially when CN range is not low (in these cases, when range $\geq$ 6m. CN connectivity increases with increasing number of people in the building and higher connectivity leads to more communication opportunities and better overall system performance. However, after the OCES reaches a high-enough connectivity level, increases in population density negatively affect the number of people that successfully evacuate due to congestion effects. As the evacuees need more time to exit the building, the fire spreads to more locations and more paths become blocked by the hazard.

IV. Conclusions and Future Work

We have proposed two spatially distributed computing systems that provide intelligent navigation services during an emergency. The first system (IES) is based on static decision nodes (DNs) that run a distributed algorithm. The IES operates as a distributed computer which adapts to changing conditions. The “best” evacuation paths are calculated collectively by the DNs using local information and communication. The second system (OCES) is composed of wireless communication nodes (CNs) that are carried by the civilians. The CNs form an opportunistic network which enables the exchange of emergency messages among them for alerting and guiding civilians during evacuation. The OCES can be viewed as a hybrid spatial computer where parts of the system are mobile (CNs) and others are static (SNs). Both systems are supported by pre-deployed sensors (SNs) that provide real-time information on the hazard. We evaluate our systems using a distributed simulation platform (DBES). Our simulation results show that the presence of the evacuation systems benefits the evacuation procedure. Our simulation study also shows how parameters such as communication range, execution and communication frequency and population density, affect system performance. In future work we will study the performance of our systems when failures are present, as well as mechanisms which can improve their performance under such conditions.

References

[1] T. J. Shields and K. E. Boyce, “A study of evacuation from large retail stores,” *Fire Safety Journal*, vol. 35, no. 1, pp. 25 – 49, 2000.

[2] M. Batalin, G. S. Sukhatme, and M. Hattig, “Mobile robot navigation using a sensor network,” in *IEEE International Conference on Robotics and Automation*, April 2004, pp. 636–642.

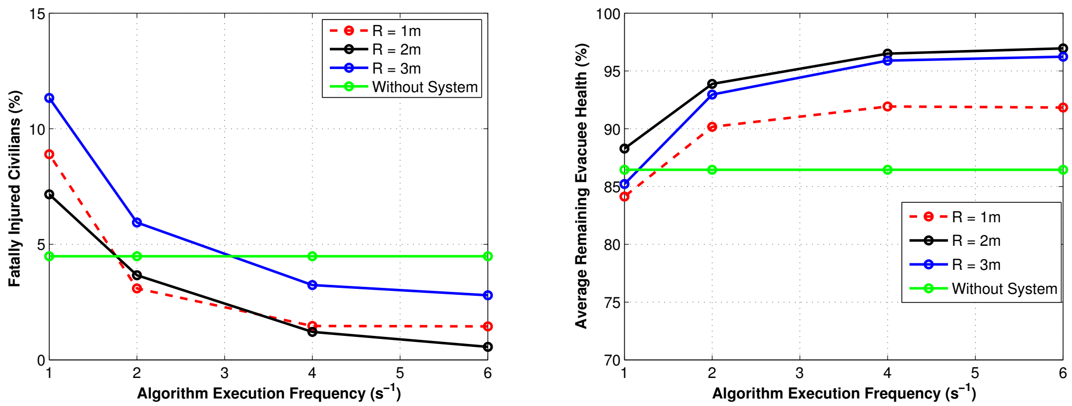
[3] P. Corke, R. Peterson, and D. Rus, “Networked robots: Flying robot navigation using a sensor net,” in *11th International Symposium of Robotics Research (ISRR 2003)*. Springer-Verlag, October 2003, pp. 234–243.

[4] Q. Li, M. D. Rosa, and D. Rus, “Distributed algorithms for guiding navigation across a sensor network,” in *MobiCom ’03: Proceedings of the 9th annual international conference on Mobile computing and networking*. New York, NY, USA: ACM, 2003, pp. 313–325.

[5] Y.-C. Tseng, M.-S. Pan, and Y.-Y. Tsai, “Wireless sensor networks for emergency navigation,” *Computer*, vol. 39, no. 7, pp. 55–62, 2006.

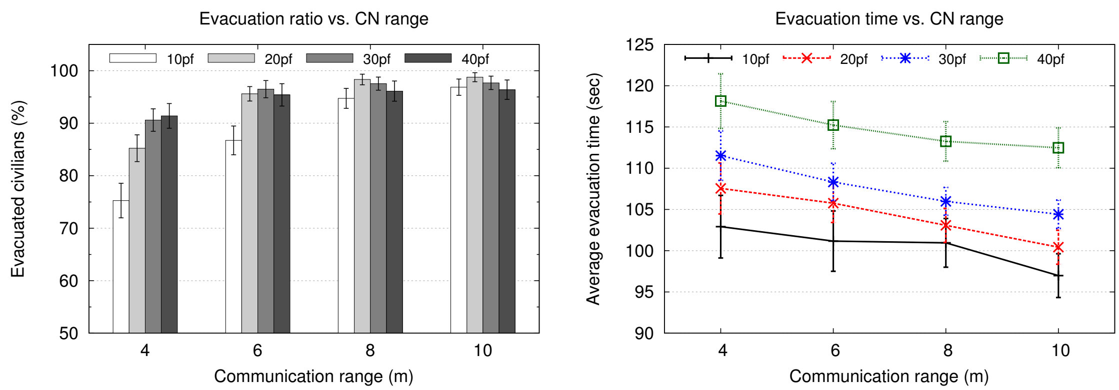
[6] A. Filippoupolitis and E. Gelenbe, “A distributed decision support system for building evacuation,” in *Proceedings of the 2nd IEEE International Conference on Human System Interaction*. Catania, Italy: IEEE, New York, NY, USA, May 21-23 2009, pp. 323–330.

[7] D. G. Elms, A. H. Buchanan, and J. W. Dusing, “Modeling fire spread in buildings,” *Fire Technology*, vol. 20, no. 1, pp. 11–19, 1984.



(a) (b)

Figure 2. Simulation results for the IES, with population density of 20 people per floor and different execution frequencies and SN ranges (R)



(a) (b)

Figure 3. Simulation results for the OCES, with different population densities and CN ranges

[8] A. M. Hasofer and D. O. Odigie, “Stochastic modelling for occupant safety in a building fire,” *Fire Safety Journal*, vol. 36, pp. 269–289, 2001.

[9] E. Gelenbe, E. Seref, and Z. Xu, “Simulation with learning agents,” *Proc. of the IEEE*, vol. 89, no. 2, pp. 148–157, Feb. 2001.

[10] E. Gelenbe, “Sensible decisions based on QoS,” *Computational Management Science*, vol. 1, no. 1, pp. 1–14, 2004.

[11] P. A. Humblet, “Another adaptive distributed shortest path algorithm,” *IEEE Transactions on Communications*, vol. 39, no. 6, pp. 995–1003, Jun. 1991.

[12] E. Gelenbe, R. Lent, and Z. Xu, “Measurement and performance of a cognitive packet network,” *Computer Networks*, vol. 37, no. 6, pp. 691–701, Dec. 2001.

[13] L. Pelusi, A. Passarella, and M. Conti, “Opportunistic networking: Data forwarding in disconnected mobile ad hoc networks,” *IEEE Communications Magazine*, vol. 44, no. 11, pp. 134–141, Nov. 2006.

[14] A. Vahdat and D. Becker, “Epidemic routing for partially-connected ad hoc networks,” Duke Univ. Dept. of Computer Science, Technical Report CS-2000-06, Apr. 2000.

[15] L. Song and D. F. Kotz, “Evaluating opportunistic routing protocols with large realistic contact traces,” in *Proc. 2nd ACM Workshop on Challenged Networks (CHANTS’07)*. ACM, 2007, pp. 35–42.

[16] A. Filippoupolitis, L. Hey, G. Loukas, E. Gelenbe, and S. Timotheou, “Emergency response simulation using wireless sensor networks,” in *Ambi-Sys ’08: Proceedings of the 1st international conference on Ambient media and systems*, 2008.

[17] N. Dimakis, A. Filippoupolitis, and E. Gelenbe, “Distributed building evacuation simulator for smart emergency management,” *The Computer Journal*, vol. 53, no. 9, pp. 1384–1400, 2010.