

A Dynamic Model for Fire Emergency Evacuation Based on Wireless Sensor Networks

Tatiana Tabirca^{1,2}, Kenneth N. Brown² and Cormac J. Sreenan¹

¹ Mobile and Internet System Laboratory and ²Cork Constraint Computation Centre,
Department of Computer Science
University College Cork, Ireland
{tabirca1, k.brown, cjs}@cs.ucc.ie

Abstract— This work introduces a dynamic model for the fire emergency evacuation problem. The model extends the concept safety introduced by Barnes et.al. for the situation when the navigation graph is dynamic. The two possible scenarios are described for using the dynamic model with a Wireless Sensor Network for fire emergency evacuation.

Keywords-*Dynamic Networks, Sensors, Safety, Fire-Fighter Assistance.*

I. INTRODUCTION

In fire safety engineering, assisted emergency navigation through indoor environments is a current research topic [17], [19]. Recent research [2], [7], [12] or [13] proposes active sensors (wired or wireless) which monitor the environment to detect variation in temperature and smoke. The sensors can also be used to provide instruction for evacuees based on the data acquired by each node and on the topology of the network, deriving the safest route to take in order to reach the exit. The decision can be taken locally by each active sensor based on only the information provided by all its neighbors if it is equipped with some computing resource. The decision can be also taken centrally at a base station (BS) node. In this case the sensing data must be gathered from all sensors to the base station which then derives the best strategy for each possible scenario. The sensors then receive from the base station accurate information about the safest route to use. Most of these solutions consider 2D or 3D maps of building floors which are represented in planar graphs.

This work presents a new approach based on *dynamic graphs* [1] for emergency evacuation. In a dynamic graphs, some elements (e.g. nodes or arcs or arc weights) vary in time. Our model uses a dynamic graph where the arc weights change over time depending on the presence of fire hazard; the weight of an arc represents the predicted travel time along the arc and this varies depending on the time at which the arc is traversed relative to the arrival of the hazard. The hazard information from the deployed sensors is centrally used to generate the dynamic graph in which safe transitions are represented by finite weights. From this a dynamic shortest path algorithm is applied to find the shortest evacuation routes. This work also uses the concepts of hazard and safety introduced in [2] but redefines them as dynamic hazard and safety based on dynamic graphs. The dynamic safety then provides the safety evacuation path which represents the path on which

we can delay safely. The dynamic model is used in a centralized manner to generate the shortest and safety evacuation paths over time. The main advantage of this centralized computation is that the evacuation information offers a better accuracy than the distributed approaches. Hence, the work focuses on the aspects of this new dynamic model rather than various technical details connected with the WSN network.

II. RELATED WORKS

Important research has been carried out to develop optimal or efficient models for building evacuation during fires. These models are usually based on maximum flows when the number of inhabitants is large [17], [19] or on shortest distances when this number is small [18].

Zlatova and Pu [18] reviewed the main evacuation systems currently available; this allowed them to outline a list of characteristics that are important for evacuation. They investigated how the evacuation routes can be generated by applying shortest path solutions to 3D graphs representing the buildings layouts. For that they analysed various solutions from breadth-first and depth-first searches to Dijkstra's algorithm.

An important focus of this type of research is on the use of WSN technology in building evacuation. Concrete details about the technical elements of WSN networks used for emergency evacuation are given in Ramuhalli et.al.[15]. Their system uses hazard sensors for detecting fire, smoke or biological agents and actuator sensors for indicating evacuation direction; sometime these two functions are present in a single node. These nodes are called "*Distributed Event Processing Nodes (DEPN)*". The authors described the main events that take place in the WSN network from the self-configuration process to the process of assisting in emergency evacuation. No details are given about how the re-routing changes under hazard, or the impact of a spreading hazard in the system. An important contribution of this work is given by an attempt to describe theoretically the states of each DEPN node. The authors considered that the state of a node S_i at the time t should be given by its set of measurements, the responses of the neighbor actuators at time $t-1$ and the states of the neighbor sensors at time $t-1$ e.g. $Node_i(t) = (readings_i(t), Actuators(t-1), Node(t-1))$. Another important conclusion of this work is that the time of evacuation based on actuator guidance is far smaller than the time of evacuation based on crowd intuition.

Chistakos [5] proposed an evacuation model based on shortest distances in the WSN topology graph. He made an analogy between emergency evacuation of inhabitants and routing data in ad-hoc network by considering travel distances as packet hops. Chistakos modified the *Destination-Sequenced Distance Vector (DSDV)* protocol from data routing into a Geographical-DSDV protocol to suit the evacuation. Each sensor of the WSN is aware of its position on the map building and knows the positions of the neighbor sensors. Furthermore, each sensor also stores the shortest distance from itself to the nearest exit and its neighbor sensor on this path. Hence, the information about the shortest path in the WSN network is distributed amongst the sensors. When a sensor node becomes hazardous then it sends notification to all the sensors in its vicinity that it cannot longer be used for safe evacuation. Then the sensors recalculate the new shortest path by requiring data from all the sensors in the vicinity and updating the shortest distance information.

Barnes et.al. [2] proposed a solution for fire emergency evacuation based on graphs which can incorporate 3D cases naturally without using specialized nodes for exits or isolated floors. The model monitors the dynamics of fire and the progress of evacuation to ensure that evacuees stay safely ahead of the hazard. This is done by using two weighed graphs over the WSN structure which are called “navigation graph” and “hazard graph”. The nodes of these graphs are all the sensors of the network. The weights are defined by:

- Hazard weight $F_{u,v}$ is the time taken by fire to spread from sensor u to sensor v .
- Navigation weight $R_{u,v}$ is the time taken for a human to travel from u to v .

Suppose that at a given moment the WSN detects some fire hazard locations so there are some sensors in hazardous area. In this case a hazard time H_u can be associated with each node in the graphs as follows

$$H_u = \min \left\{ \sum_{i=1}^{n-1} F_{u_i, u_{i+1}} : (u_0, \dots, u) \text{ with } u_0 \text{ hazardous} \right\}.$$

which can be calculated by some graph traversal. The safety of a path can be calculated using these elements recursively using the following rules:

- $S((v_{exit})) = H_{v_{exit}}$ is the hazard time to reach exit.
- Suppose that the safety of the path $p = (v_1, v_2, v_3, \dots, v_{exit})$ is know then the path $p_1 = (v_0, v_1, v_2, v_3, \dots, v_{exit})$ has the safety $S(p_1) = \min \{S(p) - R_{v_0, v_1}, H_{v_0}\}$.

The safety of a node is then defined as the maximum of safeties of all the paths from the node to exit. Barnes et.al. [2] showed how this safety scheme is implemented in WSN so that each sensor can either detect fire becoming hazardous or can update its information about the hazard time and then calculate the new safety time. Experiments proved the scheme works well delivering useful information for evacuation [2]. The authors also outlined

several drawbacks of this scheme that include possible inaccuracy of the evacuation information.

III. THE DYNAMIC MODEL

We assume that the number of people to evacuate is not too big to exceed the corridor capacities. In this case the evacuees can take any route without meeting congestions or crowded access corridors or access points. We also assume that the sensing network is deployed efficiently in the building so that it can detect fire hazards in early stages. The network can be made with wired sensors or with wireless sensors or with combination of both wired and wireless sensors. We also assume that the network has a good degree of coverage (details are given in [12] or [16]) so that all the rooms, corridors or stairs have sensors deployed.

The dynamic model starts from two static graphs. Firstly, the navigation graph $G = (V, A)$ reflects the layout of the building where $V = \{u_1, u_2, \dots, u_n\}$ is the set of chosen sensors and $A = \{(u, v) : u \text{ and } v \text{ are connected}\}$ is the set of arcs; the navigation arcs are paths in the building that we can walk. Secondly, the hazard graph is $G' = (V, A')$ which has the same nodes and $A' = \{(u, v) : \text{the fire hazard can spread from } u \text{ to } v\}$. The following weights are considered to be known for the graphs (for simplicity we work with the same notations as in Barnes et.al.[2]).

- Hazard weight or time $F : A' \rightarrow [0, \infty)$, $F(u, v) =$ the time taken for the fire to spread from u to v with $(u, v) \in A'$.
- Navigation weight or time $R : A \rightarrow [0, \infty)$, $R(u, v) =$ the time taken for an evacuee to move from u to v with $(u, v) \in A$, when there is no hazard.

For simplicity, we assume that there is only one exit node e in the navigation graph.

The dynamic model is analyzed over a time interval $Time = \{0, 1, 2, \dots, Tmax\}$ where the time unit can be for example one second. The maximum time $Tmax$ can be infinite or can be fixed to a value we want to work with. Alternatively, we can consider the time when the exit node e becomes hazardous, after which there are no safe evacuation routes.

Suppose that the nodes $\{u_1, u_2, \dots, u_q\}$ detect fire at time 0.

The first concept to introduce is the function fire $fire : A \rightarrow \{0, 1, \dots\}$ defined by $fire(u) = t$ if t is the estimated time when the sensor u detects fire. This function satisfies the following equations:

$$\bullet \quad fire(u_1) = fire(u_2) = \dots = fire(u_q) = 0. \quad (1.a)$$

$$\bullet \quad fire(u) = t \text{ if } \exists v \in A \text{ with } fire(v) < t, (v, u) \in A \text{ and } t = fire(v) + F(v, u) \quad (1.b)$$

The second rule (1.b) says that $fire(u) = t$ when there is a neighbor sensor node v that detected the fire before at the time $fire(v) < t$ and $t = fire(v) + F(v, u)$. The function $fire$ can be generated by using a simple traversal of the graphs. Initially, we start from the nodes $\{u_1, u_2, \dots, u_q\}$ whose $fire$ values are 0. Then for each node v which has a value for

fire, we consider all its neighbors u ; if the node u has not been visited yet then its *fire* becomes $fire(v) + F(v, u)$; otherwise if $fire(v) + F(v, u)$ is smaller than $fire(u)$ then this time is updated to $fire(v) + F(v, u)$.

The next concept to introduce is the dynamic graph $G^{(t)} = ((V, A), c^{(t)})$ which has $G = (V, A)$ as the underlying graph and $c^{(t)}$ as dynamic cost for the arcs defined by Equation (2). In this equation the estimated times $t_1 = \min\{fire(u), fire(v)\}$, $t_2 = \max\{fire(u), fire(v)\}$ represent the times when the fire reaches nodes u and v .

$$c^{(t)}(u, v) = \begin{cases} R(u, v) & \text{when } t < \min\{fire(u), fire(v)\} \\ R(u, v) + f(u, v; t - t_1), & t_1 \leq t \leq t_1 + time(t_1, t_2) \\ \infty, & t > t_1 + time(t_1, t_2) \end{cases} \quad (2)$$

Firstly, if there is no hazard along the arc (u, v) then the cost should be as the navigation weight. The second case considers the arc (u, v) under hazard; the hazard is detected at the time $t_1 = fire(u)$ and travels along the arc until the time $t_2 = fire(v) > t_1$. We consider that in few seconds after t_1 the arc (u, v) is totally unusable in both directions as the hazard can take over the navigation path hence $c^{(t)}(u, v) = \infty$. We also assume that the arc (u, v) can be useable from u to v for the first few moments of time e.g. $t_1, t_1 + 1, \dots, t_1 + time(t_1, t_2)$ after the hazard starts. However, the navigation weights must increase as we should factorize the time to avoid various obstacles like small fire, smoke etc. This increase depends on various factors but for simplicity we consider that depends on the arc to use (u, v) and on the length of time $t - t_1$. For that we can define a function $f(u, v; t)$ representing an amount of time with which the weight $R(u, v)$ increases after t second of hazard and a function $time(t_1, t_2) = t_3$ that gives the number of times to increment the weights.

The hazard and navigation weights, which are used in the static graphs are predetermined by some simulation or by some experiments. Olenick et.al [11] surveyed the main offline modeling and simulation methods for fire safety analysis. The 'Detector Response Model (DRM)' model of fire dynamic can be used to generate the hazard weights [11], [2].

For the arc (u, v) the hazard weight $F(u, v)$ can be considered as the fastest time the hazard moves along the arc. These values are provided by offline hazard simulations and experiments, or from best conservative estimates of fire safety engineers. On the other hand the navigation weights represent the maximum amount of time a person can walk the arc. These times are usually provided by various offline simulators for evacuation [11].

A Hazard, Shortest Path and Safety Path

This section defines the concepts of dynamic hazard and safety values as well as the shortest and safety paths for a node.

Definition 1. The dynamic hazard function is defined as follows

$$H^{(t)} : V \rightarrow R, H^{(t)}(u) = fire(u) - t. \quad (3)$$

$H^{(t)}(u)$ represents the estimated time left for the node u to become hazardous at the time t . It is clear that the hazard value is negative if the node is already under hazard and positive when the node has $H^{(t)}(u)$ time to be under hazard.

Based on the dynamic graph $G^{(t)} = ((V, A), c^{(t)})$ we can consider the dynamic shortest paths between two nodes. The cost of the path $P = (u_0, u_1, \dots, u_p)$ starting from u_0 at the moment t to u_p is

$$C^{(t)}(P) = c^{(t)}(u_0, u_1) + \dots + c^{(t_{p-1})}(u_{p-1}, u_p) \quad (4)$$

where $t_1 = t + c^{(t)}(u_0, u_1), \dots, t_{p-1} = t_{p-2} + c^{(t_{p-2})}(u_{p-2}, u_{p-1})$.

The dynamic shortest path between u_0 and u_p at the time t is the path that minimizes the cost (4) hence we can speak about the dynamic shortest path cost given by

$$D^{(t)}(u_0, u_p) = \min\{C^{(t)}(P) : P \text{ is path between } u_0, u_p\} \quad (5)$$

Definition 2. The shortest path $ShortestPath^{(t)}$ is the dynamic shortest path in $G^{(t)}$ from a node u to the exit e . $D^{(t)}(u)$ is cost of the shortest path from a node u to the exit e at time t .

One can note the shortest path $ShortestPath^{(t)}$ always uses safe arcs when $D^{(t)}(u) < \infty$. This path can be used in evacuation by fire-fighters or well able evacuees to reach the exit node in the fastest time possible. When $D^{(t)}(u) = \infty$ there are no safe evacuation paths from the node u . Chabani [3,4] outlined several dynamic shortest path problems that can be studied and developed simple solutions to solve them. An important case for the shortest dynamic path problem is when the dynamic costs $c^{(t)}(u, v)$ satisfy the FIFO rule

$$t_1 < t_2 \Rightarrow t_1 + c^{(t_1)}(u, v) < t_2 + c^{(t_2)}(u, v). \quad (6)$$

This rule gives that a person travelling on the arc (u, v) at the time t_1 arrives at v no later than a person travelling at time $t_2 < t_1$.

Proposition 1. The cost function $c^{(t)}(u, v), t = 0, 1, \dots$ from Equation (2) satisfies the FIFO rule.

For the FIFO case Chabani [3] showed that the values $D^{(t)}(u)$ can be calculated with Equation (7)

$$D^{(t)}(u) = \begin{cases} \min_{u, v \in A} \{c^{(t)}(u, v) + D^{(t+c^{(t)}(u, v))}(v)\} & u \neq e \\ 0, & u = e \end{cases} \quad (7)$$

which can be translated in a simple algorithm (Listing 1).

The algorithms to calculate the values $D^{(t)}(u)$ developed by Chabani [3] are simple and intuitive with a small complexity if the value of $Tmax$ is small. Moreover, they generate information about the evacuation paths at any

time. However, more complex solutions for the dynamic shortest path problems can be found in [1], [6] or [14].

```

// the initialisation step
for t=0 to Tmax-1 do
     $D^{(t)}(e) = 0$  ;
    for each v in  $V \setminus \{e\}$  do
         $D^{(t)}(u) = \infty$  ;  $Successor^{(t)}(u) = u$ ;

// find the static shortest path at Tmax
StaticShortestPath(  $D^{(Tmax)}$  ,  $Successor^{(Tmax)}$  ) ;
// iterate backward
for t = Tmax-1 downto 0 do
    for each node u in A do
        for each node v adjacent to u do
            if  $D^{(t)}(u) > c^{(t)}(u,v) + D^{(t+c^{(t)}(u,v))}(v)$  then
                 $D^{(t)}(u) = c^{(t)}(u,v) + D^{(t+c^{(t)}(u,v))}(v)$  ;
                 $Successor^{(t)}(u) = v$ ;

```

Listing 1. Algorithm for $D^{(t)}(u)$ in the FIFO case.

Barnes et.al. [2] introduced the concept of safety which uses the following equation $S(u) = \min\{H(u), \max\{S(w) - R(u,w) : (u,w) \in A\}\}$ to calculate the static safety values. We extend this equation to define the dynamic safety as in Definition 3.

Definition 3. The safety values $S^{(t)}(u), u \in V$ at time t are defined by

$$S^{(t)}(u) = H^{(t)}(u) \text{ when } u = e \quad (8.a)$$

$$S^{(t)}(u) = \min \left\{ H^{(t)}(u), \max_{(u,w) \in A} \left\{ S^{(t+c^{(t)}(u,w))}(w) - c^{(t)}(u,w) \right\} \right\} \text{ when } u \neq e. \quad (8.b)$$

If w^* is the node that achieves the maximum in Equation (8) then the maximum safety path or simply the safety path from u to e at time t is recursively defined by

$$SafetyPath^{(t)}(u) = \begin{cases} (u, SafetyPath^{(t+c^{(t)}(u,w^*))}(w^*)) & u \neq e \\ (u), u = e \end{cases}$$

The path $SafetyPath^{(t)}(u)$ allows fire-fighters and people in the building to delay along the evacuation path to exit. This path can be very useful for fire-fighters assisting injured people to evacuate as it provides maximum safety. Definition (3) gives the maximum amount of time one can delay the departure from u and still find a safe evacuation route to the exit e .

When u is the exit node then the safety value is equal to the hazard. Suppose the safety of u is to be calculated. Firstly, all the neighbors w should be considered. The node w is reached along the arc (u,w) at the time $t + c^{(t)}(u,w)$ so that its safety is $S^{(t+c^{(t)}(u,w))}(w)$. Hence,

we can delay in u for $S^{(t+c^{(t)}(u,w))}(w) - c^{(t)}(u,w)$ if the arc (u,w) is used; then we choose the arc (u,w) that maximizes these quantities. On the other hand the departure from u must take place before u becomes hazardous hence the definition from Equation (8.b).

The safety values $S^{(Tmax)}(u), u \in V$ are calculated statically by using

$$S^{(Tmax)}(u) = \min \left\{ H^{(Tmax)}(u), \max \left\{ S^{(Tmax)}(w) - c^{(Tmax)}(u,w) \right\} \right\}$$

in the same way as in Barnes et.al.[2]. It is clear that the safety of the exit is the same as the hazard e.g. $S^{(Tmax)}(e) = H^{(Tmax)}(e)$.

Proposition 2. The value $S^{(t)}(u)$ satisfies the following properties:

$$t \leq t' \Rightarrow S^{(t)}(u) \geq S^{(t')}(u) \quad (9.a)$$

$$S^{(t)}(u) - t' > S^{(t+t')}(u). \quad (9.b)$$

These equations establish that the safety values are decreasing over time and can be demonstrated by induction.

Proposition 3. The value $S^{(t)}(u)$ represents the maximum amount of time one can safely delay at node u .

Proof. Suppose that one can delay $t' > S^{(t)}(u)$ more than the safety at the node u so that there is a path between u and e at time $t+t'$. This path uses only arcs of finite costs hence $S^{(t+t')}(u) \geq 0$. Now Equation (9.b) is used to

provide $0 = S^{(t)}(u) - S^{(t)}(u) > S^{(t+S^{(t)}(u))}(u)$. But the safety values are decreasing so that we can find

$$\begin{aligned} t' > S^{(t)}(u) &\Rightarrow t + t' > t + S^{(t)}(u) \Rightarrow \\ &\Rightarrow 0 \leq S^{(t+t')}(u) \leq S^{(t+S^{(t)}(u))}(u) < 0 \end{aligned}$$

which is a contradiction.

These recursive equations can be simply translated into the code of Listing 2. The safety values at time $Tmax$ are computed statically. Then based on Equation (8) the safety values are computed downwards for $t=Tmax-1, \dots, 1, 0$ (See Listing 2).

```

// the initialisation step
for t=0 to Tmax-1 do
     $S^{(t)}(e) = H^{(t)}(e)$  ;  $Successor^{(t)}(e) = -1$ ;
    for each u in  $V \setminus \{e\}$  do
         $S^{(t)}(u) = -\infty$  ;
         $Successor^{(t)}(u) = e$ ;

// find the static safety at time Tmax
StaticSafetyPath(  $S^{(Tmax)}$  ,  $Successor^{(Tmax)}$  ) ;
// iterate backward
for t = Tmax-1 downto 0 do
    for each node u in A do
        for each node w adjacent to u do
            if  $S^{(t)}(u) < S^{(t+c^{(t)}(u,w))}(w) - c^{(t)}(u,w)$  then
                 $S^{(t)}(u) = S^{(t+c^{(t)}(u,w))}(w) - c^{(t)}(u,w)$  ;
                 $Successor^{(t)}(u) = w$ ;
 $S^{(t)}(u) = \min \{ S^{(t)}(u), H^{(t)}(u) \}$ ;

```

Listing 2. Algorithm to Calculate the Dynamic Safety

In conclusion this dynamic model provides useful information to evacuees and fire-fighters represented by:

- $H^{(t)}(u)$ - the time left for the node u to become hazardous.
- $S^{(t)}(u)$ - the time one can delay safely at node u .
- $D^{(t)}(u)$ - the quickest time to reach the exit on safe arcs.
- Two evacuation routes given by either the shortest path $ShortestPath^{(t)}(u)$ or by the safety path $SafetyPath^{(t)}(u)$.

Proposition 4. The evacuation information represented by the values $S^{(t)}(u)$ and $D^{(t)}(u)$ is calculated in $O(m + n \cdot \log n + T \max(n, m))$.

Proof. The algorithms presented above in Listing 2 and 3 have similar structure. Firstly, they initialize $D^{(t)}(u)$, $S^{(t)}(u)$ and $Successor^{(t)}(u)$ respectively in $n \cdot Tmax$ operations. Secondly, the static functions $StaticShortestPath$ and $StaticSafetyPath$ are called. $StaticShortestPath$ can be in fact any All-to-One algorithm for shortest path; our implementation used the Dijkstra algorithm which has the best complexity $O(m + n \cdot \log n)$ when a Fibonacci heap data structure is used [1]. On the other hand $StaticSafetyPath$ provides a computation for the safety values $S^{(Tmax)}(u), u \in V$ so that $S^{(Tmax)}(u) = \min\{H^{(Tmax)}(u), \max\{S^{(Tmax)}(w) - c^{(Tmax)}(u, w)\}\}$.

These values can be calculated based on a computation similar to the Dijkstra algorithm in which the Fibonacci heap data structure is used to provide efficiently the maximum value [1]. Hence, we can consider that the complexity of $StaticSafetyPath$ is still $O(m + n \cdot \log n)$. Finally, the algorithms retrospectively compute the required values for $t = Tmax - 1, \dots, 1, 0$. For each step t the algorithms go through the nodes and through the arcs adjacent to each node. Hence, this traversal process requires a number of $2 \cdot m$ operations for each time t . We can conclude that the overall complexity is $O(m + n \cdot \log n + T \max(n, m))$.

IV. TWO SCENARIOS TO USE THE DYNAMIC MODEL

In this section we present two possible scenarios in which the dynamic evacuation model can be applied. Both of them use the base station resources to simulate the dynamic evacuation model.

Scenario 1. *The WSN network detects the fire at some locations and notifies the base station. Then the base station computes the evacuation routes with the dynamic evacuation model. Finally, the base station sends the evacuation information through the WSN network.*

This scenario is feasible under the assumption that the probability of having the base station under the fire hazard is theoretically zero. The architecture of WSN network for this scenario uses a topology with two backbones. Firstly, we have backbone formed with hazard sensors that

monitor constantly the environment to detect variations in temperature, smoke or gases. Secondly, there is a backbone formed with special actuator sensors which are deployed to provide visual and perhaps audio information to evacuees about the safety route to follow. These actuator sensors receive data from the base station about the safety time and about the successor node in the safety path for a sequence of times. Then the actuator sensors display visually the direction to follow. These specialized sensors can be located on corridors, intersections, stairs etc. Sometime, the actuators can be mobile, being deployed on robots that can assist in evacuation. The sensors of the two backbones form the hazard and navigation graphs. For simplicity the two graphs are considered to be undirected so that an arc (u, v) can be traversed in both directions with the same navigation costs. A room can have two or multiple doors with the same corridor or with different corridors so that it can be possible that the safety path to go through the room. As the actuator sensors are located on corridors any evacuation path has to contain at least a few of them.

Scenario 2. *The WSN network detects the fire at some locations and notifies the base station and the Incident Commander (IC) point too. Then the base station or the IC point computes the evacuation routes with the dynamic evacuation model. Finally, the base station and the IC notify the evacuation leaders and the fire-fighters respectively about evacuation routes.*

The first two stages of this scenario are identical to the ones in Scenario 1. The difference is that the base station or the IC point communicates with fire-fighters. The fire-fighters are equipped with various sensing and communication gadgets including Head Mounted Display (HMD) and have direct communication to the IC point (see [20]). Moreover, the fire-fighters whereabouts are always known at the IC point. Hence, the IC can send the evacuation information about the hazard and safety times and the successor of the node where the fire-fighter is located. The fire-fighter can choose between either the shortest path or the safety path. The HMD device can then relay this information to the fire-fighter in a visual or audio way. This is a model in which the evacuation scenarios are computed centrally at the base station or at the IC point. In this way the simulation results are accurate and provide estimated information about the evacuation routes. Using a separate backbone for the actuator sensors enables a quick communication to take place between the base station and the actuator nodes.

V. EVALUATION OF THE SCENARIOS

The evaluation of this system takes into consideration the response time of each stage of the above schemes. Firstly, the sensing WSN network detects the fire hazard and sends the information to the base station. This information is packed in a package which usually contains the sensing data, the sensor location etc. The package traverses the WSN to the base station which is notified about the hazard. The time that a package reaches the base station depends on the data routing protocol that is used in WSN network. Ikikardes [9] investigated the routing

solutions in WSN network for safe critical applications. He developed a new type of dynamic routing called “N SafeLink” which is low-latency, robust and energy efficient. *Ikikardes et.al. [10]* found that the response time must be of maximum $\tau_{fire} = 10s$ for a message to travel from a sensor that detects fire to the BS node in the WSN network.

The second stage is at the base station where the computation of the evacuation information takes place. The fast computation of the evacuation over some time is paramount in this case. It is clear that the execution time depends of the number of nodes and arcs in the navigation graph as well as on the maximum time T_{max} as Proposition 4 illustrates.

This execution time does not depend however on the number of initial hazard locations. For the evaluation we consider that the sensors in the hazard and navigation graphs are placed in a squared grid fashion with nr rows and columns. The exit is always at the right bottom node of the grid. This graph is very rich in elements with nr^2 nodes and $2 \cdot nr \cdot (nr - 1)$ arcs. The hazard and navigation weights are randomly generated for simplicity. Starting from this initial information the dynamic model computes the *fire* values and the hazard function and then it generates the dynamic weights. Using these dynamic weights, the model then computes the dynamic shortest paths and values and the safety path and values. The dynamic model was implemented in C++ and its execution used a standard laptop machine with a Pentium 2.0 MHz processor and a RAM memory of 2.5 GB. Firstly, the execution of the dynamic model is observed for all the values of $nr=11, 13, 15, 17, 19, 21$. The value of T_{max} is chosen to be $fire(e)$ so that the evaluation is carried out until the exit node becomes hazardous. One can see that the execution times for less than 225 rooms is negligible being less than 0.5 second. However, the execution for 289, 361 and 441 rooms exceeded one second where the evaluation was done over more than 200 time units (see Table 1).

Rooms	121	169	225	289	361	441
Corridors	264	364	480	612	760	924
T_{max}	119	149	174	204	223	244
Time (s)	0.08	0.18	0.48	1.05	2.6	5.1

Table 1. Execution Times for $T_{max}=fire(e)$.

The evaluation can be run up to the first 100 time units in order to keep the execution times under one second. Table 2 proves that these execution times can be reduced to nearly one second if the time is up to 100 seconds. A comparison between the execution times for $T_{max}=fire(e)$ and $T_{max} = \min\{fire(e), 100\}$ is illustrated in Figure 1.

However, the execution times become bigger as the number of rooms increases. Even when the evaluation is done for a limited time these values are still in the order of a few seconds.

Rooms	121	169	225	289	361	441
Corridors	264	364	480	612	760	924
T_{max}	100	100	100	100	100	100
Time (s)	0.08	0.13	0.37	0.53	0.83	1.34

Table 2. Execution Times for $T_{max}=\min\{fire(e), 100\}$.



Figure 1. Execution Times for $T_{max}=fire(e)$ vs $T_{max}=\min\{fire(e), 100\}$.

In this case the dynamic algorithm can be ineffective and more advanced methods must be employed. A possible choice to reduce the execution times is to spread the execution over the grid (see the FireGrid project [21]) or to use HPC computing. *Therefore, we can conclude that the computation of the dynamic algorithm for 100 time units can be done in less in 2 seconds at the base station for floors with less than 400 rooms.*

The second backbone of the WSN network contains the actuator sensors used to raise the alarm and to display the evacuation information. The solution introduced by *Ikikardes [9]* considers a maximum of 6 hops in the topology with a maximum 1 second communication between hops. *Hence, the sensor actuator network can display the evacuation information received from the BS node in maximum 6 seconds.*

The communication between the IC point and each fire-fighter and between the base station and evacuation leaders take place through direct channels. A detailed study of this communication through the IEEE 802 Wireless Technologies is outlined by *Hofmann et.al. [8]*. Several parameters of the communication were analyzed under various types of hazard. The main conclusion was that the wireless communication technologies in the band of 2.4 GHz are suitable for fire-fighters. Moreover, the communication is not affected by temperature or smoke hazards but this can be instead affected by vapors. The other important information of this study is that the latency of communication is up to 150 milliseconds when the high quality voice over is transmitted. Similar considerations can be made if evacuation leader have mobile devices enabled with IEEE 802 Wireless communication technology. *Hence, the packages with evacuation information can reach the fire-fighters or evacuation leaders in real time with virtually nil latency.*

To conclude the centralized system can provide evacuation information to evacuees on the actuator sensors within 17 seconds from the moment when a sensor detects the hazard. The evacuation leaders can have the evacuation routes on their devices on 11 seconds. Moreover, the fire-fighters are provided in real time with the accurate evacuation information.

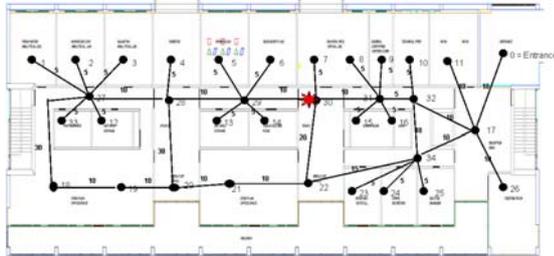


Figure 2: Evacuation Example.

A Practical Example

We now consider an example based on the ground floor of the UCC Environmental Research Institute building. This floor's rooms and corridors can be modeled with 35 nodes and 39 arcs (see Figure 2) for the navigation graph. The navigation weights were generated by considering the physical distance between two nodes and the slowest time a person can traverse the arc. For example the navigation weight between node 1 and node 27 is 5 seconds while the navigation weight between 27 and 28 is 10 seconds.

The hazard graph uses the same nodes as in the navigation graph; the hazard arcs are formed with the arcs from Figure 2 plus some arcs formed by neighbor rooms e.g. (1,2), (2,3), (3,4), (4,5), (5,6), (6,7), (7,8), (8,9), (9,10), (10,11), (33,12), (23, 24), (24,25). Each hazard weight represents an estimation of how quick the fire hazard can spread between 2 nodes. For example the hazard weight between node 1 and node 27 is 30 seconds while the hazard weight between 1 and 2 is 60 seconds. The initial location of the fire hazard is supposed to be at node 30. We assume that the arc (u,v) under hazard at time t can be still usable for up to 5 more seconds and the arc's costs increase with 2 for each extra second.

The model introduced above provides the dynamic hazard, shortest path cost and safety values as information about the fire hazard. The hazard values are decreasing on each second while the shortest path cost and safety change when the dynamic cost changes. After the hazard starts at node 30, the node's adjacent arcs change their values; for example the navigation weights of the arc (30, 31) are 10, 12, 14, 16, 18, 20 and then ∞ for the subsequent times. Node 7 has the shortest path to the exit node 0 as (7, 30, 31, 32, 17, 0) with the cost 50 while for $t=1$ the shortest path changes to (7, 30, 22, 34, 17, 0) with the cost 88. Note that for $t \geq 2$ there is no safe path from node 7 as we reach node 30 at time $t=9$ or more when the costs of the arcs adjacent with 30 are all ∞ .

The computation gives that the safety of node 7 is -10 for $t=0$ and -17 for $t=1$ which means that there is not time to delay in node 7 but there is still a safe path to the exit 0. Moreover, the algorithm provides a safety path which is

similar to the shortest path. For $t > 1$ the safety of node 7 becomes $-\infty$ and there is no safe path from the node to the exit. This is because all the arcs adjacent to 7 are not usable from the 7th second on (see Figure 3).

Now let's follow the shortest and safety paths from node 29 change at $t=0,1$ (Figure 4). Figure 4 illustrates the shortest path with red line and the safety path with blue line at times 0 and 1. At time 0 the shortest path from 29 is (29, 30, 31, 32, 17, 0) with the cost 50. For $t=1$ the shortest path changes to (29, 28, 20, 21, 22, 34, 17, 0) with the cost 70. The safety path for $t=0$ and $t=1$ is (29, 28, 20, 21, 22, 34, 17, 0) with the safety values of 20 and 19 respectively. Finally, we can mention that the execution time of this practical example was 0.003 seconds for $T_{max}=250$ seconds.

VI FINAL CONCLUSIONS

This paper presented a new approach based on dynamic graphs for the emergency evacuation problems. The dynamic graph was defined so that the arcs' weights change over time to reflect the dynamics of the fire hazard. The dynamic hazard and safety metrics were also introduced to measure the amount of time until a node becomes hazardous and the time a person can safely delay at a node, respectively. Two dynamic algorithms were described to generate the dynamic shortest paths and the dynamic safety paths. The dynamic shortest path can be used by able evacuees to reach the exit in the fastest time possible. On the other hand the dynamic safety path can be used by injured evacuees or by fire-fighters assisting them in order to have a good margin of delay.

Then the paper proposed two scenarios in which the dynamic model can work with a WSN network for fire emergency problems. The performance evaluation showed that the solution can be effective for emergency evacuation. Our simulations showed that the computation time can be under a second for building environments with less than 400 rooms with evacuation information generated for up to 100 seconds.

This work will be further extended by considering several possible improvements. Firstly, maximum flow will be considered in the dynamic model in order to avoid door, corridor or stair congestions. Secondly, the first scenario will be generalized for the case when multiple base stations are available. More research will be needed for the case of multiple base station data routing in the fire emergency.

ACKNOWLEDGMENTS

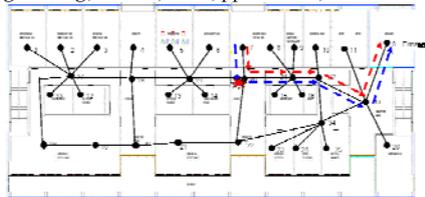
This research is part of the *NEMBES* project which is funded by the Irish Higher Education Authority under their PRTLIV Research Programme.

REFERENCES

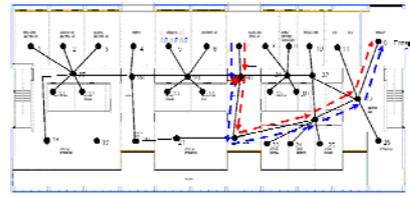
- [1] R. K. Ahuja, K. Ravindra, J.B. Orlin, S. Pallottino, and M. Scutella. "Dynamic Shortest Paths Minimizing Travel Times and Costs", *MIT Sloan Working Paper No. 4390-02*, 2002.
- [2] M. Barnes, H. Leather and D.K. Arvind, "Emergency Evacuation using Wireless Sensor Networks", *Proceedings of the 32nd IEEE*

Conference on Local Computer Networks (LCN 2007), pp 851-857, 2007.

- [3] I. Chabini, "A new Shortest Path Algorithm for Discrete Dynamic Networks", *Proceedings of the 8th IFAC Symposium on Transport Systems*, Chania, Greece, June 16-17, pp.551-556, 1997.
- [4] I. Chabini, "Discrete Dynamic Shortest Path Problems in Transportation Applications: Complexity and Algorithms with Optimal Run Time", *Transportation Research Record* 1645, pp.170-175, 1998.
- [5] C. Christakos: Sensor Networks Applied to the Problem of Building Evacuation: An Evaluation in Simulation, *Proceedings of the 15th IST Mobile and Wireless Summit*, pp. 245-258, June, 2006.
- [6] C. Demetrescu and G. F. Italiano: "Dynamic Shortest Paths and Transitive Closure: Algorithmic Techniques and Data Structures", *Journal of Discrete Algorithms*, Vol 4, Issue 3, pp. 353-383, 2006.
- [7] A. Filippopolitis, L. Hey, G. Loukas, E. Gelenbe and S. Timotheou: "Emergency Response Using Wireless Sensor Networks", *e-Proceedings of the ICAMS Conference*, Quebec City, Canada, pp. February 2008.
- [8] P. Hofmann, K. Kuladinithi, A. TimmGiel, C. Görg, C. Bettstetter, F.Capman, C. Toulosaly, "Are IEEE 802 Wireless Technologies Suited for Fire Fighters?," Whitepaper, *wearIT@work project*, EU, 2006.
- [9] T. Ikkardes: "Routing Algorithms for Safety Critical WSN Networks", *MSc Thesis, SS-2006*, Swiss Federal Institute of Technology Zurich, 2006.
- [10] T. Ikkardes, M. Hofbauer, A. Kaelin, M. May: "A Robust, Responsive, Distributed Tree-Based Routing Algorithm Guaranteeing N Valid Links per Node", *Wireless Ad-Hoc Networks. ISCC 2007*, pp. 455-462, 2007
- [11] S. M. Olenick and D.J.Carpenter, "An updated international survey of computer models for fire and smoke," *SFPE Journal of Fire Protection Engineering*, vol. 13, no. 2, pp. 87-110, 2003.
- [12] M.S. Pan, C. H. Tsai and Y.C. Tseng, "Emergency Guiding and Monitoring Applications in indoor 3D Environments by Wireless Sensor Networks", *Int. J. Sensor Networks*, Vol. 1, Nos. 1/2, pp. 2-10, 2006.
- [13] S.Potter and G. Wickler: "Model-Based Query Systems for Emergency Response", *Proceedings of the ISCRAM 2008 Conference*, pp., 2008
- [14] G. Ramalingam and Thomas W. Reps: "An Incremental Algorithm for a Generalization of the Shortest-Path", *J. Algorithms*, 21(2), pp. 267-305, 1996.
- [15] P. Ramuhalli and S. Biswas: "Managed Traffic Evacuation Using Distributed Sensor Processing", *Proceedings of the SPIE*, Volume 5769, pp. 48-58 2005.
- [16] A. Tsertou, R. Upadhyay, S. McLaughlin and D. Laurenson "Towards a Tailored Sensor Network for Fire Emergency Monitoring in Large Buildings", Digital Repository Infrastructure Vision for European Research, 2007-2008.
- [17] P. G. Torre, B.A. Díaz, V. Ordóñez and J.J. del Coz, "A Model for Defining Evacuation Policies for Emergency Escape from Buildings", *Journal of Simulation and Multidisciplinary Design Optimization* 2, pp. 237-244 2008.
- [18] S. Zlatova and S. Pu, "Evacuation Route Calculation of Inner Buildings", Research Book Chapter in *Geo-Information for Disaster Management*, Springer, pp. 1143-1161, 2005.
- [19] P. Wang, P. B. Luh, S.C. Chang and J. Sun, "Modeling and Optimization of Crowd Guidance for Building Emergency Evacuation", *Handbook of Intelligent Robotics and Applications*, LNCS 5315/2008, pp 1-6, 2008.
- [20] J. Wilson, V. Bhargava, A. Redfern, P. Wright, "A Wireless Sensor Network and Incident Command Interface for Urban Firefighting", *Mobile and Ubiquitous Systems: Networking & Services*, MobiQuitous, pp. 1-7, 2007.
- [21] FireGrid Project: <http://www.firegrid.org>.

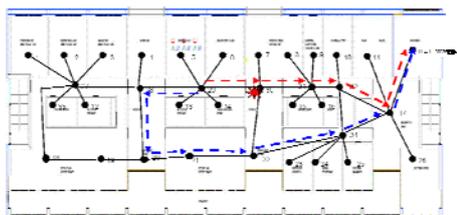


T=0

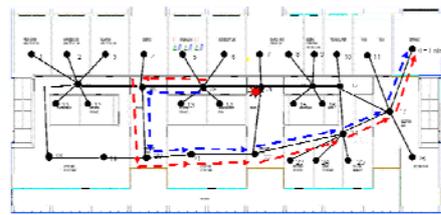


T=1

Figure 3: Dynamics of the shortest and safety path of node 7 for t=0,1.



T=0



T=1

Figure 4: Dynamics of the shortest and safety path of node 29 for t=0,1.