

An IoT-based Infrastructure to Enhance Self-Evacuations in Natural Hazardous Events

José Mariano Finochietto · Matias Micheletto · Gabriel M. Eggly ·
Roger Pueyo Centelles · Rodrigo Santos* · Sergio F. Ochoa · Roc
Meseguer · Javier Orozco

Received: date / Accepted: date

Abstract Regardless of the extensive research conducted on large scale evacuations, the instrumentation of these processes still represents an open issue for first response organizations. Self-evacuation of civilians that follow evacuation plans has shown to be feasible as early response to several natural disasters; however, the typical lack of interaction capability of the evacuees with first response organizations and emergency managers, jeopardizes the effectiveness of these processes. This article proposes an IoT-based infrastructure that supports self-evacuation of civilians in mass processes, allowing people to participate as information providers and consumers. This infrastructure is the backbone of an ambient intelligence system used as a bridge between evacuees, first response units and the emergency operation

center managing the process. Depending on the people location and the status of the area, the system implements breadcrumbs that guide people to shelters and safe places. The proposed infrastructure includes: (1) an architecture that captures the core design aspects of the solution and makes it reusable for other researchers, (2) an implementation of the system based on Raspberry Pi 3 devices with LoRa radio connectivity, (3) a mobile application that allows evacuees to interact with the evacuation system, and (4) a simplified algorithm to support the deployment of the IoT-based infrastructure into an urban area. The solution was evaluated using real measurements and simulations, and the obtained results are highly encouraging.

Keywords IoT-based infrastructure · ambient intelligence system · large scale self-evacuations · participation of civilians · human sensors · natural disasters

Gabriel Eggly, Matias Micheletto, Rodrigo Santos* and Javier Orozco
Department of Electrical and Computers -ICIC
Universidad Nacional del Sur- CONICET
Bahía Blanca 8000, Argentina
E-mail: gmeggly@gmail.com, matias.micheletto@uns.edu.ar, ierms@criba.edu.ar*, jadorozco@gmail.com

Mariano Finochietto
GIDI, Department of Information Technology
Universidad Nacional de Mar del Plata
Mar del Plata 7600, Argentina
SpinalCom, 91400 Orsay, France
E-mail: mariano.fino@fi.mdp.edu.ar

Sergio F. Ochoa
Department of Computer Science
Universidad de Chile
8370456 Santiago, Chile
E-mail: sochoa@dcc.uchile.cl

Roger Pueyo Centelles and Roc Meseguer
Department of Computer Architecture
Universitat Politècnica de Catalunya
08034 Barcelona, Spain
E-mail: rpueyo@ac.upc.edu, meseguer@ac.upc.edu

Conflict of Interest

On behalf of all the authors, the corresponding author states that there is no conflict of interest.

1 Introduction

According to the United Nations, during 2019 55.3% of the world's population lived in urban settlements. This number will increase to 60% by 2030, where one out of three people will live in cities with at least half a million inhabitants [26]. The growth of the urban population makes the cities more vulnerable to natural and man-made disasters [22]; therefore, new systems are required to make the cities more resilient.

The most accepted strategy to reach such a goal is to design robust and smart systems that allow reducing risks and mitigating the impact of extreme events on the civil population. Typically, the design and implementation of these systems are performed during the disaster “preparedness” stage (i.e., during planning of how to respond to an extreme event), and the resulting systems are used during the “response” stage [22].

Although all disaster relief activities are important, this work focuses on supporting the evacuation of civil population during natural disasters. The goal of the evacuation process is to get people to a safe place as soon as possible, considering the current and future state of the extreme event (e.g., a tsunami, hurricane, landslide, flood or wildfires).

While first responders (i.e., firemen, police officers and paramedics) usually support evacuations of civilians during urban emergencies, for example, fires in buildings. In the case of natural disasters the process is performed mainly through self-evacuations, since they are massive and conducted in parallel. Therefore, civilians and the public infrastructure should be prepared to address such a process. Counting on appropriate evacuation plans and supporting systems reduces risks and also the impact of an extreme event.

The suitability of a system to support an evacuation is a consequence of its design. Although the system implementation also plays a key role in the effectiveness of the solution, its design is much more relevant since a poor design usually cannot be fixed with an appropriate implementation.

The design of an evacuation system must be addressed at the preparedness stage, and its potential effectiveness should be evaluated theoretically and empirically to ensure its suitability when used in real scenarios. In this sense, the evaluation of the system design before implementing it, is as important as its modeling.

The design of these systems should consider the participation of evacuees as users, in addition to the regular ones, e.g., first responders and emergency managers, since the former are direct producers and consumers of information about the evacuation process and extreme event. Therefore, these systems should be interactive and hopefully provide ambient intelligence (AmI) through distributed and autonomous solutions [9].

Given that regular communication infrastructure in the affected area is usually down or collapsed during a disaster [21], the evacuation system should count on its own communication infrastructure, that should be resilient enough as to operate under disaster conditions. Considering the large number of civilians participating in these evacuations, the information flow allowed among participants should be properly designed to avoid

communication overload and the collapse of the infrastructure.

This article proposes an IoT-based communication infrastructure that allows the active participation of civilians in large scale self-evacuations. We describe its structure, behavior, the way to deploy it in urban areas, and also the way in which several user types employ it to provide and consume information. The infrastructure provides ambient intelligence to evacuees in the route towards the shelters.

The article also presents a formal representation of such an infrastructure using IoT calculus [17]. This formalization eases the early evaluation of evacuation system design and allows its improvement during the emergency preparedness process.

This paper extends a conference presentation in which the authors introduce the basic concepts of this proposal [9]. This extension includes a complete definition of the IoT-based infrastructure, its validation, a simplified strategy to deployment it, a real implementation of the infrastructure nodes and simulations real measurements that show the potential performance of a particular evacuation system.

Next section discusses the previous work on ICT-based systems to support large scale evacuations, and the proposals to model several aspects of these systems. Section 3 presents the proposed IoT-based infrastructure and its main components, including the supported roles and their communication behavior. Section 4 introduces the formalism used to represent the infrastructure and validate its operation. Section 5 presents a proof-of-concept implementation of the proposed infrastructure and the algorithm to deploy it in an urban scenario. Section 6 describes the experiments performed to validate the proposal, and the obtained results. Section 7 presents the conclusions and the future work.

2 Related Work

Designing and implementing ICT-based emergency evacuation systems represents a challenge for any organization; particularly, when these systems have to support self-evacuations. This challenge increases with the size and complexity of the area to be addressed, and also with the number and type of participants involved in the evacuation process. Thus, modeling and implementing these systems tends to be more feasible when they have to support self-evacuations from buildings than addressing mass processes (e.g., after a tsunami warning).

The advances in Internet of Things (IoT) technology have allowed us to conceive and implement ambient intelligence in confined spaces (e.g., buildings, shopping

malls and airports), and thus to better support self-evacuations processes [5]. Examples of these IoT-based evacuation systems are presented in [2, 18]. Typically, these systems determine evacuation paths in real-time, considering the characteristics of the hazard, evacuees' behavior, and environmental conditions, and then guide the evacuees to safe zones. Although useful, most IoT-based evacuation systems involve important instrumentation of the physical infrastructure, which limits their capability of addressing large scale evacuations.

In mass evacuations, early warning systems (EWS) typically use broadcast communication infrastructure (e.g., TV and radio broadcast, and siren systems) to start the evacuation process and give general indications to the evacuees. The transmission of information through these infrastructures is one-way, which restricts the participation of the evacuees as potential information providers [24].

There are also EWS and emergency management systems that allow full-duplex digital communication among first response organizations and government agencies during evacuations; e.g., through cellular and private networks [20, 29]. However, these systems are usually not available for civilians [31, 15].

Several researchers have raised the need to include the evacuees as formal actors in disaster relief efforts, given that first response organizations are usually not able to deal with these activities in short time periods [25, 36]. Therefore, civilians acting as human-sensors can provide information about the status and evolution of both, the extreme event and the evacuation process [4, 27, 34, 7]. This opens several opportunities to conceive new interactive systems, based on the Internet of People paradigm [6], that help increase the effectiveness of mass evacuations.

Concerning the participation of civilians in emergency situations, the literature reports several experiences where people use social networking services (SNS) to help deal with these events [37, 13]. However, in natural disasters the incidence area is usually affected by a communication blackout due to physical damages, traffic overload, or lack of electricity [3, 20, 29]. Therefore, the use of SNS and regular communication infrastructure is diminished or not feasible in these locations. People located outside the incidence area frequently spread information through SNS about the response process and the extreme event; however, such information tends to be outdated, inaccurate or incomplete [10]. This limits the capability of using SNS to monitor the evolution of the evacuation process and the extreme event.

The literature also reports few proposals that allow civilians located into the affected area to report and receive information during an evacuation; most of

them consider mobile ad hoc communications systems (e.g., based on Twitter or ad hoc text messages [12, 32]). These interactive systems are useful for coordinating activities in small areas, but are limited to supporting large scale evacuations since the communication infrastructure they use has low reliability and bandwidth, and short communications range [33]. In this sense, the work closest to this proposal is the communication system proposed in [30], where civilians use a mobile application and a LoRaWAN infrastructure to exchange short text messages with family members and first response organizations. Thus, these people inform their status and receive information from others. Such a system was conceived to support civilians after an earthquake, where the evacuations are not frequent and the mobility of people is low or null. Therefore, that proposal needs to be adapted for using it as communication support during large scale evacuations.

The previous works highlight the need for evacuation systems be interactive and allow ubiquitous participation of evacuees, not only as human sensors, but also as recipients of first response information. These systems capabilities must be designed and evaluated during the preparedness stage of these evacuations in order to try ensuring the effectiveness of these systems when used in practice. The formal modeling appears as an appropriate approach to represent both, the structure and behavior of the systems, and also their interaction with the potential users. This formal modeling also allows us to envision the systems capabilities, and perform early evaluations and adjustments during the preparedness process.

The literature reports several proposals to model different aspects of an evacuation system. For instance, in [19] the authors present an ontology named SEMA4A (Simple EMergency Alert 4 [for] All) to model the accessibility to emergency notification messages through distinct emergency response information systems. In [28], the authors extend previous work and introduce a semi-automatic technique for knowledge acquisition and modeling on accessible evacuation routes. Finally, in [34] the authors propose some guidelines to implement mobile applications that support emergency notifications and responses activities where citizens act as human sensors.

On the other hand, in [1] the authors report a survey on approaches for hierarchical modeling of complex IoT systems, like those required to support large evacuations. In [16] a model-driven approach is introduced to model the different users (e.g., evacuees and first responders) that participate in an AmI system. In [11] the authors introduce the calculus to represent the dynamic of these systems, based on the π -calculus pro-

posed in [23]. Following a similar approach, in [17] the authors extend the π -calculus to cope with the special characteristics of IoT devices, processes and networking services.

In this paper, the authors use the latter approach to formally represent the structure and behavior of an IoT-based infrastructure that supports self-evacuations in mass process. This formalization uses IoT-calculus to describe the system components and the interaction among them; it is also employed to validate the functionality of a particular evacuation system. Next sections describe and formalize the proposed IoT-based infrastructure.

3 Infrastructure of the Evacuation Support System

In order to understand the support provided by this infrastructure, we must first identify the types of users that employ the evacuation support system to perform their activities. These users are decision makers, first responders and evacuees as shown in Figure 1, with particular emphasis in the latter population because it is the largest one and usually not included in other ICT-based evacuation systems.

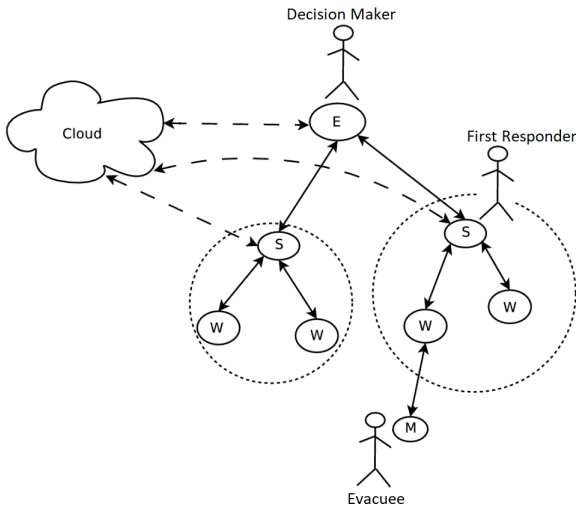


Fig. 1 Hierarchy of roles considered in the system

The proposed infrastructure does not consider direct interactions among these users, as a way to reduce the information overload on its nodes and communication links. Instead, it provides ambient intelligence through a hierarchy of components that cooperates to provide services to the users. The interaction between a user and the evacuation system, and also among components of the infrastructure, is based on a set of pre-

defined services. These services use coded structured information to ensure its appropriate transmission and processing, event if the load in the system components is high.

The participation of end-users is materialized through the use of software applications that are able to interact with particular type of node of the proposed infrastructure. The nodes types were defined as *roles* of the system components, and arranged in a hierarchy according to the proposed IoT-based infrastructure. These roles are the following: Emergency Operation Center (E), Shelters (S), Witness Units (W) and Mobile Units (M). They produce and consume information according to their type, and all of them are stationary, except the mobile units. Figure 1 presents a diagram of the infrastructure that includes the hierarchy of roles and the interactions allowed among them.

The emergency operation center is the root of the system; it concentrates all the available information and also most of its processing. Emergency managers monitor and coordinate the evacuation efforts from this node. The shelters are managed by first responders that receive, host and eventually route the evacuees. Every shelter reports on its status (i.e., their availability and capability to receive evacuees) to the emergency operation center and witness units located within its influence area. The witness units are autonomous components that allow mobile units (i.e., the evacuees) to access the evacuation support infrastructure, e.g., to report on their location or to get suggested evacuation routes to reach the closest shelter. Finally, the mobile units are represented by the mobile devices used by the evacuees.

These nodes can produce and consume different types of information according to their hierarchy, and these actions can be done either by an explicit order of a human being (user), or as result of unattended interactions among the autonomous agents running into the nodes. The type of interactions allowed among nodes should be defined at the system design time, since it affects the system scalability, its capability to deal with the information flow, and consequently, its potential effectiveness to support large scale evacuations.

This infrastructure may operate independently of counting on a connection to Internet, however, a copy of its information is kept on the cloud as backup, to increase the information availability in case of collapse of nodes or communication links, and also to allow the remote monitoring of the process without interfering with the system operation. Although nodes type E and S are prepared to work under extreme conditions, a satellite connection is available between these nodes and the cloud (dotted links in Figure 1).

3.1 Roles Description

Each node type (i.e., role considered in the infrastructure) has particular capabilities to produce and consume information, and also to interact with other nodes. Next, we describe the properties and capabilities of these nodes.

3.1.1 Mobile Units

The nodes labeled as M represent the mobile devices used by the evacuees (e.g., a smartphone). These devices must have a WiFi interface and a specific software application that allows users to connect to W nodes (i.e., witness units) near to the users location. Once launched, the application scans the environment looking for a W that serves as access points to log in to the evacuation system. This allows the M nodes to retrieve updated evacuation routes based on their location using the information exchanged with the W. The mobile units can report on their current location and also the presence of obstacles in the area through the witness unit they are connected to.

3.1.2 Witness Units

As mentioned, the Ws allow Ms to access the same information available in the system; e.g., the evacuation routes to get the closest shelter. These nodes can be seen as access points that at the same time are end-nodes of an LPWAN (Low Power Wide Area Network). The functionality of the Ws allows them to receive information about new obstacles from Ms, compute the routes based on the information of the area, and send back an updated route to the nearest safe shelter. The number of devices connected to a W allows the system to envision the number of evacuees that require a shelter in the area, and the amount of people using the different paths. The interaction between Ms and Ws should be lightweight (e.g., through coded JSON messages) to increase as much as possible the capability of Ws to act as an access point for the evacuees.

The coverage that provides by the Ws in the evacuation area strongly influences the capability of the AmI system to route evacuees toward shelters. Therefore, the deployment of these nodes should be planned carefully to ensure as much coverage as possible in the evacuation area.

3.1.3 Shelters

The shelters, labeled as S, are the destination of the evacuees. The features of each shelter (e.g., location,

capability, availability, occupancy and influence area) are informed and updated by the first response teams in charge of managing these nodes. This information is periodically delivered through the system since it has direct consequences in the evolution of the evacuation process.

Typically, a shelter is represented in the system through one or more computers with satellite connection to the cloud and wireless connection to both the Ws in its influence area and the emergency operation center. As shown in Figure 1, each S has a set of Ws that are distributed within its influence area. When an S is no longer available for any reason, the Ws route people to another S.

3.1.4 Emergency Operation Center

This node, labeled as E, represents the main repository and information processing unit of the evacuation system; it is also the root and sink of the proposed infrastructure. Besides receiving all the messages from the Ss, E has to update the evacuation routes towards the Ss and account for their capacity to receive new evacuees.

The software application used by decision makers to monitor the process allows them to add or remove obstacles in the evacuation area. Using such an additional input, the evacuation system can dynamically calculate more appropriate routes for the evacuees and keep informed to nodes.

3.2 Information Flow

Figure 2 shows the information flow between the roles considered in the proposed infrastructure. The information moves along the system through hierarchical paths. Information about obstacles or blockings detected by evacuees eventually reach the E in an upstream propagation, through Ws and Ss. This information is later propagated downstream to keep all nodes properly informed. The figure also shows the way in which Ws and Ss make local decisions based on data coming from Ms.

3.3 Fault Tolerance

The proposed system relies on a distributed infrastructure, where some nodes may come down due to several reasons. Therefore, its capability to recover from partial failures was considered in the infrastructure design. For instance, the nodes implement a keep alive protocol by periodically exchanging status messages as a way to detect failures in the system nodes.

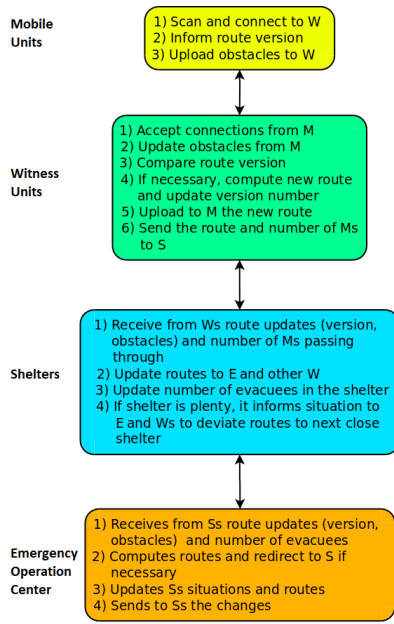


Fig. 2 Information flow between roles in the system

When a W is down, the shelter linked to that unit assumes the function of the W and informs the situation to E. When a shelter is down (or overcrowded), the Ws linked to it recognize the situation and route evacuees to the next closer shelter. The E also detects the situation and informs accordingly to the other shelters in the area.

When the E is down, the hierarchical structure of the system allows each shelter to assume the coordination of the Ws into its influence area. In consequence, several infrastructure islands are created and operate independently one of the other until the E come be available again to coordinate and unify the system. Although this last situation has a low probability of happening, since the E is particularly prepared to deal with extreme events, it was also considered in the design of the infrastructure.

4 Modeling Interactions Between Roles

In this section, we formally describe the nodes and their interactions using IoT-calculus [17]. The use of a formal representation allows us to validate the design of a particular evacuation system that uses the proposed infrastructure; particularly, it is possible to determine the capability of the system to support interactions and information flow among the nodes. Thus, the potential performance of the system can be evaluated through simulations, and its design can be adjusted at preparedness time, considering the simulation results.

Following the semantics proposed in IoT-calculus, the whole system may be described by (1):

$$n_E \leftrightarrow n_S | n_S \leftrightarrow n_W | n_W \leftrightarrow n_M \vdash \{E|S|W|M\} \quad (1)$$

where n_x indicates a node of type x . In this case, E, S, W, or M \leftrightarrow indicates the presence of a link communication between both nodes, and $|$ indicates that links, nodes or processes work in parallel.

The scope of this modeling has been limited to better describe the main interactions between the roles considered in the system. To further focus on modeling the interactions, the local computing that each node performs has not been defined in the calculus, but encapsulated in the functions below:

- **setV(v)**: Stores the route version number v in the current node.
- **getV()**: Returns the route version number stored in the current node.
- **setO(o)**: Stores the obstacles o in the current node.
- **getO()**: Returns the obstacles stored in the current node since the last time this function was invoked.
- **setR(r)**: Stores the route changes r in the current node.
- **getR(v)**: Returns the route changes since the route version number v .
- **updateR(o)**: Calculates and stores a new route and route version number, based on the current stored route, and added or removed obstacles in o .

It is assumed that every time a new route is received, conflict management and route merging is solved by the above functions.

Once the system is formally defined, the next step is to identify the processes running in each kind of node, and the channels and variables they use to communicate within a node and between different nodes.

4.1 Interactions Between Mobile and Witness Units

Provided that evacuees move towards the Ss during the evacuation process, it is necessary to represent the fact that Ms and Ws will have transitory connections between them. Typically, the Ms scan the environment looking for the presence of a W that allow them to log in to the system, and thus to get or update their evacuation route. For this reason, the Ms automatically send a broadcast request *wifiConn* with the download channel *wifiPull*, and the values of the locally stored route version v and new obstacles o . For ease of understanding, we have left out of the model the details involved in the process by which both roles get connected.

Once these nodes establish a connection, the W calculates and updates the route and its version based on

the obstacles received from the mobile units (if any), and it uses the download channel *wifiPull* to send the last route *r* and version *v*. The route *r* could be just the difference between the M known route and the newly created route in W, while the version *v* could be a number based on a timestamp related to the last route. Figure 3 shows the channels and interaction between the nodes M and W.

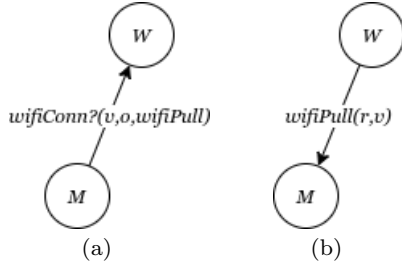


Fig. 3 (a) M sends a connection request and its local data to W, and (b) W sends the last route and version to M.

The formal definition of M can be found at (2). It is composed of two processes running in parallel. The process *P* is in charge of interacting with node W and updating the mobile display a_m , while the process *Q* interacts with a user's explicit input through s_m to get the list of obstacles in the area and record them into the system.

$$\begin{aligned}
 M &= n_M[!P \mid !Q] \\
 P &= \overline{\text{wifiConn}}!(\text{getV}(), \text{getO}(), \text{wifiPull}). \\
 &\quad \text{wifiPull}(r, v).\text{setV}(v).r \rightarrow a_m \\
 Q &= (o) \leftarrow s_m.\text{setO}(o)
 \end{aligned} \tag{2}$$

The description of the Ws is more complex as they interact with Ms, but also with Ss. The representation of their interaction is completed with Figure 4.

4.2 Interactions Between Witness Units and Shelters

Once the W has a new route, apart from updating the M, it will transmit these new changes to S through the *loraPush* channel. The S will then use such a channel to broadcast updates to other Ws (represented in the diagram as W') whose evacuation routes are affected by these changes. In this case, it is not necessary to establish a connection as it was with WiFi, since many LPWANs (such as, LoRa or LoRaWAN in ABP mode) do not require a join procedure.

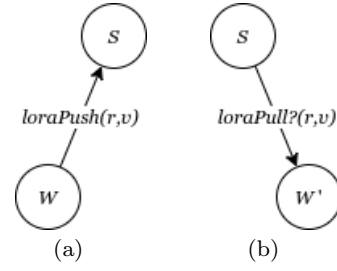


Fig. 4 (a) Once W computes a new route, it sends it to S, and (b) S updates the other corresponding witness units W' whose routes have been affected.

The node W, which is formally defined in (3), simultaneously runs processes *R* and *T*. The process *R* accepts the connection request from the node M, computes the new route and then sends the updated route data to nodes Ms and S in two parallel automatic subprocesses. The process *T* describes node W while getting updated route data from node S.

$$\begin{aligned}
 W &= n_W[!R \mid !T] \\
 R &= \overline{\text{wifiConn}}?(v, o, \text{wifiPull}).\text{updateR}(o). \\
 &\quad (\overline{\text{wifiPull}}\langle \text{getR}(v), \text{getV}() \rangle \mid \\
 &\quad \overline{\text{loraPush}}\langle \text{getR}(v), \text{getV}() \rangle) \\
 T &= \text{loraPull}?(v, r).\text{setV}(v).\text{setR}(r)
 \end{aligned} \tag{3}$$

4.3 Interactions Between Shelters and the Emergency Operation Center

The Ss are similar to Ws as they keep a double connection: one with the Ws and the other with the E. The interaction with E can be done with a reliable satellite internet connection or a reliable long-range RF protocol. Every node S can be thought as an LPWAN (e.g., LoRa or LoRaWAN) gateway. When it receives new route data, it sends it to Ws as explained before, and also to the node E with a unidirectional link named *internetPush*. Other nodes (S') will receive this update from E through a broadcast transmission channel named *internetPull*. This is illustrated in Figure 5.

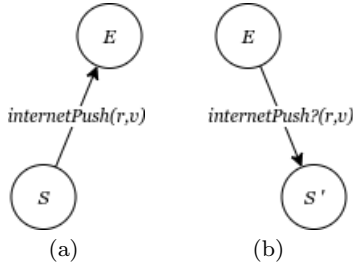


Fig. 5 (a) S sends new route data to E, and (b) E updates the other shelters S' with the received data.

S is formally defined at (4). As in previous cases, two parallel processes establish the behavior of this node type. Process U receives route data from the W and sends it to E and other Ws. Process V receives data from E and sends it to the Ws into the influence area of S.

$$\begin{aligned}
 S &= n_S[!U \mid !V] \\
 U &= \text{lorapush}(r, v). \text{setR}(r). \text{setV}(v). \\
 &\quad \overline{\text{lorapull}}!(r, v) \mid \overline{\text{internetpush}}(r, v) \\
 V &= \overline{\text{internetpull}}?(r, v). \text{setR}(r). \text{setV}(v). \\
 &\quad \overline{\text{lorapull}}!(r, v)
 \end{aligned} \tag{4}$$

Finally, the emergency operation center (E) is described at (5), in which the processes X and Y are involved. The first one stores and broadcasts to the Ss the new route data received from one S. The second one receives information about obstacles from the decision makers through s_e , then updates the route data, and finally, it broadcasts such information to the Ss.

$$\begin{aligned}
 E &= n_E[!X \mid !Y] \\
 X &= \overline{\text{internetpush}}(r, v). \text{setR}(r). \text{setV}(v). \\
 &\quad \overline{\text{internetpull}}!(r, v) \\
 Y &= (o) \leftarrow s_e. \text{updateR}(o). \overline{\text{internetpull}}!(r, v)
 \end{aligned} \tag{5}$$

4.4 Formalizing the Information Flow Through the System

In (6) we show how the system evolves from an M reporting an obstacle, until the updated route reaches E. To better understand the flow of information along the system of nodes, these aspects have been taken into account:

- As the nodes have several processes running concurrently, only those having activity are shown, while the other nodes are written as $n_x[*]$, being x the name of the node, indicating that they are not performing actions.

- Only process names have been used to abstract the details of the implementation.
- Trigger functions have been added to each process, and thus new instances can be created and executed. The name of the triggers are composed by the string *trig* and a suffix containing the name of the process.

The formalization of the information flow starts in 6(a) with the formulation of the nodes and the triggers. In the next iteration of the system -6(b)-, all the triggers are fired so each process generates a new instance and waits for data through their input channels. At 6(c) an evacuee (i.e., a M node) submits an obstacle o' through s_m . Then, M finds a W and connects to WiFi in 6(d). In 6(e) W calculates a new route (and a new version number) based on the received information. The system then iterates to 6(f), where W sends the new route and version number to M, and this latter updates the screen of the mobile application a_m in 6(g). At this point, the instance of process P generated at 6(b) finishes its execution in M.

The system continues processing and evolves to 6(h), where S receives the new route and version number from W; thus, the instance of process R in W finishes its execution. Finally, S sends the new route and version number to E in 6(i). Now, E has an updated route that was generated considering the information of an obstacle submitted by an evacuee. From this point, the system is ready to further update the information stored in all Ss and Ws participating in the evacuation system.

5 System Implementation

The proposed system was implemented as a proof-of-concept within the Digital Signal Laboratory at the Southern National University, Argentina. In this implementation, several out-of-the-shelf components were used as building blocks. Basically, the system requires components that can implement the four roles already defined: mobile unit, witness unit, shelter and the server at the emergency operation center. Next, we present a brief description of these components.

5.1 Mobile Units

These units are the regular devices used by evacuees (e.g., their smartphone), and they represent the M nodes in the system. These devices run a software application designed to interact with the infrastructure (particularly with witness units) and thus get support to the user during the evacuation.

The application shows a map of the area where the evacuee is located, and also several marks including user

$$\begin{aligned}
& \text{(a) } n_E \leftrightarrow n_S \mid n_S \leftrightarrow n_W \mid n_W \leftrightarrow n_M \vdash n_M [!trigP().P.\overline{trigP}\langle \rangle \mid !trigQ().Q.\overline{trigQ}\langle \rangle \mid \overline{trigP}\langle \rangle \mid \overline{trigQ}\langle \rangle] \mid \\
& \quad n_W [!trigR().R.\overline{trigR}\langle \rangle \mid !trigT().T.\overline{trigT}\langle \rangle \mid \overline{trigR}\langle \rangle \mid \overline{trigT}\langle \rangle] \mid \\
& \quad n_S [!trigU().U.\overline{trigU}\langle \rangle \mid !trigV().V.\overline{trigV}\langle \rangle \mid \overline{trigU}\langle \rangle \mid \overline{trigV}\langle \rangle] \mid \\
& \quad n_E [!trigX().X.\overline{trigX}\langle \rangle \mid !trigY().Y.\overline{trigY}\langle \rangle \mid \overline{trigX}\langle \rangle \mid \overline{trigY}\langle \rangle] \xrightarrow{\tau} \\
& \text{(b) } n_E \leftrightarrow n_S \mid n_S \leftrightarrow n_W \mid n_W \leftrightarrow n_M \vdash n_M [!trigP().P.\overline{trigP}\langle \rangle \mid !trigQ().Q.\overline{trigQ}\langle \rangle \mid P.\overline{trigP}\langle \rangle \mid Q.\overline{trigQ}\langle \rangle] \mid \\
& \quad n_W [!trigR().R.\overline{trigR}\langle \rangle \mid !trigT().T.\overline{trigT}\langle \rangle \mid R.\overline{trigR}\langle \rangle \mid T.\overline{trigT}\langle \rangle] \mid \\
& \quad n_S [!trigU().U.\overline{trigU}\langle \rangle \mid !trigV().V.\overline{trigV}\langle \rangle \mid U.\overline{trigU}\langle \rangle \mid V.\overline{trigV}\langle \rangle] \mid \\
& \quad n_E [!trigX().X.\overline{trigX}\langle \rangle \mid !trigY().Y.\overline{trigY}\langle \rangle \mid X.\overline{trigX}\langle \rangle \mid Y.\overline{trigY}\langle \rangle] \xrightarrow{\tau: o' \leftarrow s_m} \\
& \text{(c) } n_E \leftrightarrow n_S \mid n_S \leftrightarrow n_W \mid n_W \leftrightarrow n_M \vdash n_M [!trigP().P.\overline{trigP}\langle \rangle \mid !trigQ().Q.\overline{trigQ}\langle \rangle \mid P.\overline{trigP}\langle \rangle \mid \{o'/o\}Q.\overline{trigQ}\langle \rangle] \mid \\
& \quad n_W [*] \mid n_S [*] \mid n_E [*] \xrightarrow{\overline{wifiConn}\langle v', o', wifiPull \rangle} \\
& \text{(d) } n_E \leftrightarrow n_S \mid n_S \leftrightarrow n_W \mid n_W \leftrightarrow n_M \vdash n_M [!trigP().P.\overline{trigP}\langle \rangle \mid !trigQ().Q.\overline{trigQ}\langle \rangle \mid \{v'/getv\}\{o'/o\}P.\overline{trigP}\langle \rangle \mid \overline{trigQ}\langle \rangle] \mid \\
& \quad n_W [!trigR().R.\overline{trigR}\langle \rangle \mid !trigT().T.\overline{trigT}\langle \rangle \mid \{v'/v\}\{o'/o\}R.\overline{trigR}\langle \rangle \mid T.\overline{trigT}\langle \rangle] \mid \\
& \quad n_S [*] \mid n_E [*] \xrightarrow{\tau} \\
& \text{(e) } n_E \leftrightarrow n_S \mid n_S \leftrightarrow n_W \mid n_W \leftrightarrow n_M \vdash n_M [!trigP().P.\overline{trigP}\langle \rangle \mid !trigQ().Q.\overline{trigQ}\langle \rangle \mid \{v'/getv\}\{o'/o\}P.\overline{trigP}\langle \rangle \mid \overline{trigQ}\langle \rangle] \mid \\
& \quad n_W [!trigR().R.\overline{trigR}\langle \rangle \mid !trigT().T.\overline{trigT}\langle \rangle \mid \{v''/v\}\{o'/o\}R.\overline{trigR}\langle \rangle \mid T.\overline{trigT}\langle \rangle] \mid \\
& \quad n_S [*] \mid n_E [*] \xrightarrow{\overline{wifiPull}\langle r', v'' \rangle} \\
& \text{(f) } n_E \leftrightarrow n_S \mid n_S \leftrightarrow n_W \mid n_W \leftrightarrow n_M \vdash n_M [!trigP().P.\overline{trigP}\langle \rangle \mid !trigQ().Q.\overline{trigQ}\langle \rangle \mid \{v'/getv\}\{o'/o\}\{r'/r\}P.\overline{trigP}\langle \rangle \mid \\
& \quad \overline{trigQ}\langle \rangle] \mid \\
& \quad n_W [!trigR().R.\overline{trigR}\langle \rangle \mid !trigT().T.\overline{trigT}\langle \rangle \mid \{v''/v\}\{o'/o\}R.\overline{trigR}\langle \rangle \mid T.\overline{trigT}\langle \rangle] \mid \\
& \quad n_S [*] \mid n_E [*] \xrightarrow{\tau: r' \rightarrow a_m} \\
& \text{(g) } n_E \leftrightarrow n_S \mid n_S \leftrightarrow n_W \mid n_W \leftrightarrow n_M \vdash n_M [!trigP().P.\overline{trigP}\langle \rangle \mid !trigQ().Q.\overline{trigQ}\langle \rangle \mid \overline{trigP}\langle \rangle \mid \overline{trigQ}\langle \rangle] \mid \\
& \quad n_W [!trigR().R.\overline{trigR}\langle \rangle \mid !trigT().T.\overline{trigT}\langle \rangle \mid \{v''/v\}\{o'/o\}R.\overline{trigR}\langle \rangle \mid T.\overline{trigT}\langle \rangle] \mid \\
& \quad n_S [*] \mid n_E [*] \xrightarrow{\overline{lorPush}\langle r', v'' \rangle} \\
& \text{(h) } n_E \leftrightarrow n_S \mid n_S \leftrightarrow n_W \mid n_W \leftrightarrow n_M \vdash n_M [*] \mid \\
& \quad n_W [!trigR().R.\overline{trigR}\langle \rangle \mid !trigT().T.\overline{trigT}\langle \rangle \mid \overline{trigR}\langle \rangle \mid T.\overline{trigT}\langle \rangle] \mid \\
& \quad n_S [!trigU().U.\overline{trigU}\langle \rangle \mid !trigV().V.\overline{trigV}\langle \rangle \mid \{r'/r\}\{v''/v\}U.\overline{trigU}\langle \rangle \mid V.\overline{trigV}\langle \rangle] \mid \\
& \quad n_E [*] \xrightarrow{\overline{internetPush}\langle r', v'' \rangle} \\
& \text{(i) } n_E \leftrightarrow n_S \mid n_S \leftrightarrow n_W \mid n_W \leftrightarrow n_M \vdash n_M [*] \mid n_W [*] \mid \\
& \quad n_S [!trigU().U.\overline{trigU}\langle \rangle \mid !trigV().V.\overline{trigV}\langle \rangle \mid \{r'/r\}\{v''/v\}U.\overline{trigU}\langle \rangle \mid V.\overline{trigV}\langle \rangle] \mid \\
& \quad n_E [!trigX().X.\overline{trigX}\langle \rangle \mid !trigY().Y.\overline{trigY}\langle \rangle \mid \{r'/r\}\{v''/v\}X.\overline{trigX}\langle \rangle \mid Y.\overline{trigY}\langle \rangle] \mid
\end{aligned}$$

(6)

location, the evacuation route suggested to reach the closest shelter, and obstacles in the path, e.g., dropped electricity cables, fires and inaccessible bridges (Figure 6(a)).

Using the application, the user can act as a human sensor during the evacuation process, providing information on obstacles still not informed. Touching the event location on the device screen, the application shows a menu with regular obstacles to choose (Figure 6(b)). Once an obstacle has been added to the system, the W updates the information locally, dismissing the routes passing by that area and issuing the alert to its S and indirectly to E. The system also presents indications (distance, direction and estimated travel time) to

reach the shelter, considering the suggested evacuation route (Figure 6(c)).

In the user interfaces, the location of the Ws are shown as blue markers, the locations of danger zones are displayed as red markers with their respective icons, and the red path on the street marks the suggested path that a user should follow. Every marker on the map has a popup menu with detailed information. On the upper right section, the user interface shows a drop-down button with instructions to the evacuee.

This application uses the witness units to implement as a breadcrumb trail that guides the evacuees to the shelters. In such a process, the M nodes connect alternatively to several Ws according to the people lo-

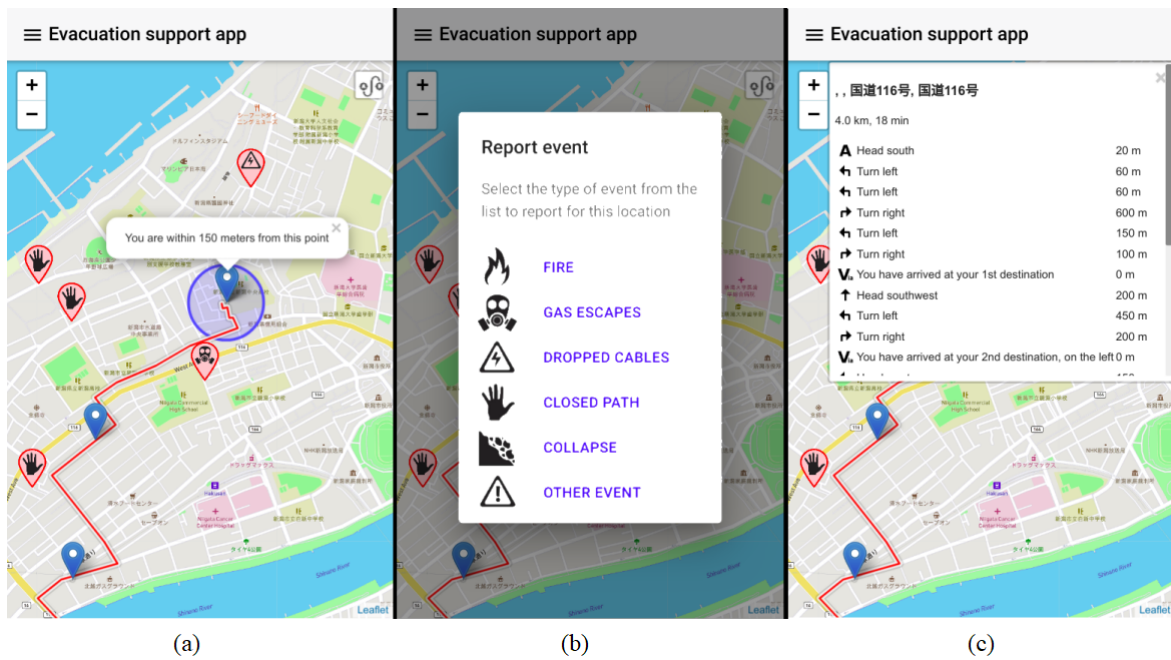


Fig. 6 User Interfaces of the Evacuation Support System: (a) evacuation route, (b) interface for obstacle report, and (c) indications to reach the shelter

cation. The actions performed by the mobile devices to keep end-users connected to the infrastructure (and also to provide them ambient intelligence) are invisible for the evacuees. Therefore, these end-users perceive the mobile application as an ubiquitous system that helps them reach a shelter, regardless their current location. In this sense, the appropriate deployment of Ws in the evacuation area plays a key role in the ubiquity perceived by the end-users during the evacuation process. Section 5.7 discusses this aspect and proposes an algorithm to perform this deployment.

The mobile application uses web-based technologies, e.g., the Leaflet library to display an interactive map with routes and markers. The GUI components are provided by the Framework 7 styles library and the communication with the Ws is handled through the Axios library, which is an application based HTTP client for browsers. This implementation allows distributing the application through many ways, without modifying the source code. A first distribution option could be to generate a standalone application for mobile and desktop computers, which could be compiled with several frameworks, such as Apache Cordova or Electron. Moreover, the application can also be hosted on a web server and distributed through HTTP, as most websites. Finally, it can also be distributed as a Progressive Web Application (PWA). A PWA is a website that can be installed as an extension of a browser, and runs autonomously using its own requirements and grants.

5.2 Witness Units

The Ws play a key role as the evacuees access the evacuation system through them. Mobile users interact directly with the Ws uploading obstacles in the route and downloading the updated route to the shelter. This node type not only acts as a relay or broker between the evacuation support system and evacuees, but also has the capability to compute alternative routes based on local information. Two communication interfaces are present in each W: (a) a LoRa link that is used to exchange information with the associated S, and (b) a WiFi that is used as access point to the Ms.

The Ws can be built using commercial Raspberry Pi 3B. A LoRa interface is implemented with the TTGO LoRa32 SX1276 board. In turn, these boards interact with each other via serial communication. Figure 7 shows an alternative to implement these witness units and their interfaces to interact with mobile units and shelters.

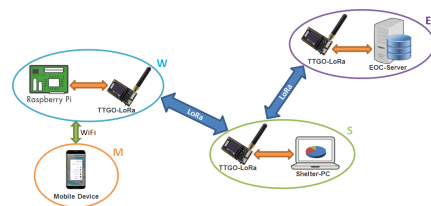


Fig. 7 System schematic

The Raspberry Pi board is configured as an access point that provides an open WiFi network under the SSID of “WU_ID”. This network is configured with a captive portal, therefore, when the mobile user accesses the WiFi network, the browser on the device automatically opens with a “customize” web page that shows an offline map centered on the current location of the user. This helps increase the sense of ubiquity perceived by the user.

A Web Socket Server running in the Raspberry Pi (i.e., in the witness unit) is in charge of managing the information exchange and the interaction with other nodes. Each obstacle reported by a user is stored in a JSON format file in the database of the mobile device. Every time a mobile unit connects to the WiFi network, a server script merges the local and remote databases through an information synchronization process. Such a process is not demanding in terms of computing and use of memory, since the information to be synchronized is structured, mainly based on text representations, and the synchronization algorithms are simple.

All new reports coming from the mobile nodes are queued in the W for transmitting them to the corresponding S. Therefore, the Ws act as a bridge between the Ms and the corresponding S. Moreover, all messages from nearby Ws, which arrive via LoRa communication, are relayed to the Raspberry Pi server script via serial messages, and they are stored in the local database of the witness unit.

5.3 Shelters

The node representing a shelter (S) is usually implemented using a desktop computer with the capability of processing and storing the information of the evacuation process in its influence area. Since these units are located in safe areas, they are prepared to deal with extreme events; therefore, we can assume that S nodes will count on power and communication services during the evacuation process (or most part of it).

In the current implementation, the Ss receive information from Ws in their influence area, and upload such information to the Emergency Operation Center (E) through a LoRa link. Shelters may also have access to Internet to backup the information in the cloud while the service is available, and share that information with the S when the E is not available.

This node type periodically informs its availability to receive evacuees. When its capacity limit is reached, it informs the situation to Ws and the E as to deviate further evacuees to the nearest available shelter. The Ss also implement a keep alive protocol to detect falls in the links with the Ws and the E.

5.4 Emergency Operation Center

Similar to the Ss, the E is located in a safe area, therefore, it can be implemented using a server with multiple communication capabilities. Although it is not mandatory, this server should count on a satellite link to the Internet that allows this node to backup the evacuation information into the cloud.

This node counts on a software system that lets emergency managers establish and modify routes when required, and transmit such information to Ss and Ws. The system allows also these people to monitor the evacuation process and make decisions that are transmitted to the nodes depending on the role they play in the proposed infrastructure.

5.5 Information Availability

The strategies used to represent and share information through the IoT-based infrastructure affect the information availability; and therefore, the decision making activities and the effectiveness of the evacuation processes. For enhancing the information availability, the proposed infrastructure uses coded messages (i.e., predefined message types) structured in JSON format. This helps reduce the message size, and also understand and synchronize its content when it arrives to a destination node. Consequently, it reduces the consumption of energy and bandwidth involved in the information dissemination.

The interaction among nodes follows the hierarchy of roles shown in Figure 1, and it is performed using a master/slave paradigm, where only one node is able to transmit messages in a certain time slot. This strategy helps avoid collisions.

In order to ensure fairness on the opportunities to transmit messages, E initiates the poll in a round-robin fashion, by giving control to the first S (see Figure 8). This node acknowledges the request and indicates into the message how many Ws are working in its influence area, and also the number of evacuees hosted in the shelter. Then, the S starts the polling of the Ws also using round-robin, and each W answers with one or more messages depending on the number of new events or obstacles to report.

The answer of a W is a JSON structured object that contains three fields: the node id, the event data, and the termination message (true/false). The event data field is an array that can contain up to 5 events (a payload of 220 bytes) and the amount of Ms going through that W. Each event is represented by its location (latitude and longitude), an identifier that represents the kind of event or obstacle, a key to save it in the database

and a timestamp that indicates the moment at which it was recorded. The W filters out when more than one M upload the same event.

The last message between a W and a S includes a termination field that closes the interaction between these nodes. Once the S receives this ending message, it uses round-robin to determine the next W to poll. When all W s have been polled, the S sends the updated information to the E , and close the communication. After this, the E starts the interaction with the next S , and repeats the process.

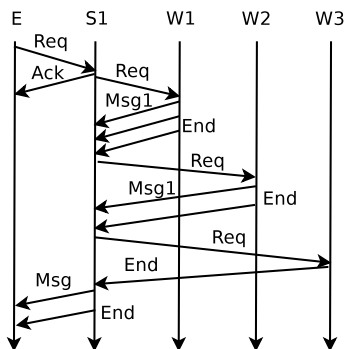


Fig. 8 Information sharing process

In evacuation scenarios, the information availability is also related to the capability of the support system to cover the physical area where the evacuees are located. Therefore, the communication technology used to implement the links among nodes, and also the algorithm used to deploy the W s, play a key role. Next sections address these two aspects.

5.6 Communication Technology

Concerning the communication links between nodes E , S and W , they were implemented using LoRa technology, and the communication between W and M nodes was implemented using WiFi. The LoRa transceivers were configured to have the largest possible communication range in urban environments, and thus to allow evacuees easy access to the information in the evacuation route. Particularly, the LoRa network was set with the following parameters: the spreading factor (SF) is set to 12, the bandwidth (BW) at 125 KHz, and the coding rate (CR) to 5. This configuration tends to minimize the message lost among nodes.

The implementation of the proposed infrastructure involves the use of simple components that are available in the local market, are reliable and their cost is affordable. These features make the proposal feasible to

be implemented in underdeveloped or developing countries. Moreover, the decision of using mobile devices (particularly, smartphones) to log in to the support infrastructure allows evacuees to have an ubiquitous and mass access to relevant information during evacuations, and contributes to the system adoption by civilians.

Concerning the deployment of the system, and considering this is a NP-hard problem, next section describes the algorithm to deploy the nodes of the proposed infrastructure.

5.7 Infrastructure Deployment

The infrastructure was designed to help provide ambient intelligence to the evacuation environment. Therefore, it should provide an effective support for the evacuees to count with a large amount of W s, conveniently deployed, to guide and route the evacuation process towards the S s. For this, the W s location must allow the system to implement breadcrumb trails that can be easily followed by evacuees while moving towards safe areas. How many W s to deploy and where to place them is a typical coverage problem that has been proved to be NP-Hard [14].

In [35] a greedy solution was introduced to address such a problem, and in [9] a meta-heuristic approach based on a genetic algorithm was proposed. The general idea for deploying the W s is to place the units in such a way that an evacuee (i.e., a mobile unit) has a maximum coverage distance D between any two of them. Thus, when a M moves from a W to the next in its evacuation route, the time elapsed without AmI support is minimum.

The distance can be measured in meters or in the average time required by an evacuee to move between the two W s. The appropriate way of measuring distance depends on the kind of evacuation being supported, since it is not the same if the M is walking or using a vehicle (scooters, cars, trucks), and also if it is moving up in a hill or in a flat territory. As distance is a variable that depends on the conditions surrounding the city on which the AmI system will be deployed, we just refer to it in an abstract way (without a unit of measure).

In order to determine a suitable number of witness units and their locations to maximize the coverage of the AmI system, the city with an area A is divided in regions $\rho = R_1, \dots, R_n$, such that $\forall i \neq j R_i \cap R_j = \phi$ and $\cup_{\forall i} R_i = A$. Usually, cities have a large street network that helps reduce these problems to smaller regions. In each region, a graph $G(\nu, \xi)$ should be built, where each node $x_i \in \nu$ represents an street intersection and the edges ξ_{ij} indicate the block linking x_i and x_j .

The weight $w(\xi_{ij})$ represents the distance, $\xi_{ij} = \xi_{ji}$. Finally, a maximum desirable distance between two Ws, D , should be defined.

The Ws location algorithm is quite straightforward. The maximum distance D must be set as parameter. Considering this value and the area of the deployment region, a regular grid can be calculated in order to achieve an equidistant distribution of Ws. The nodes in the map that are close to each point of the regular grid are selected as locations for Ws, and thus, it is possible to obtain a tentative distribution of Ws in the region.

This solution can be improved if this distribution is used as initial condition of an optimization method, as the one proposed in [9]. However, the simplicity of this deployment algorithm represents a strength.

The algorithm was run to determine the infrastructure deployment in Bahia Blanca city, in Argentina. The shape of the land makes this city prone to floods. Moreover, the city hosts a large petrochemical industrial area and a harbor where a LPG (Liquefied petroleum gas) transfer station is operating with carriers. Therefore, an evacuation support system is required for the city.

The map of Bahia Blanca city was downloaded from openstreetmap.org, and the street model was extracted from the OSM data, and converted to a fully connected graph using Octave tools. Figure 9 shows the original map of the city on the left and its graph representation on the right side. D was set to 500m.

Figure 10 shows the locations where the Ws should be deployed according to the proposed algorithm. The black dots indicate the locations of the Ws. Red circles show a possible clustering of Ws, and the location of Ss are indicated with a green dot. About 40 Ws are in the influence area of a S. As can be seen, in some cases the circles (i.e., the Ss influence area) are overlapped; however, in that situation the Ws are linked to only one S.

The distribution of Ws over the city map does not represent a perfect grid, since the street pattern in some areas is not regular. In these cases, the distribution of minimum distances between neighboring Ws results in a normal distribution, as shown in Figure 11.

6 System Evaluation

The system evaluation involved two steps: the experimental data collection and the system simulation. Next sections explain the activities involved in these steps and the obtained results.

6.1 Experimental Data Collection

The experimental data collection involved the deployment of a node representing a shelter, which was located into a place at 30m height. Such a node was linked to a witness unit using a LoRa link. The LoRa transceivers were configured as indicated in Section 5.5, i.e., with SF=12 and BW=125KHz.

The witness unit was deployed in six different points of the city, mixing residential and commercial areas, and also considering different distances between the shelter and the witness unit. In all cases the witness unit was installed in a location 1.6m height. Table 1 presents the results obtained for the RSSI (Received Signal Strength Indicator) and SNR (Signal to Noise) variables.

| Distance [m] | RSSI [dBm] | SNR [dB] |
|--------------|------------|----------|
| 2000 | -113 | -12.95 |
| 1800 | -119 | -10.50 |
| 1300 | -116 | -14.80 |
| 1000 | -111 | -10.53 |
| 700 | -118 | -9.55 |
| 500 | -115 | -16.50 |

Table 1 Values of RSSI and SNR at different distances between the shelter and the witness unit

For each location of the witness unit, five experiments were performed to report 1, 5, 10, 15, and 20 events, representing 11 messages in total. The last request from the shelter was answered with 4 messages to provide the 20 events. In all cases, the transmission delay for the link was 33.01s, with an effective throughput of 26.9 [Payload's Bytes]/[s]. The total miss ratio for the requests was 9%. These numbers illustrate the capability of the network links to deliver messages with a low miss ratio.

In order to analyze the potential performance of an evacuation support system based on the proposed infrastructure, we simulated the deployment of 100 witness units in the urban area of Bahia Blanca city. The values obtained through the experimental data collection were used to model the network infrastructure, and the simulations were implemented using Octave [8].

Concerning the suitability of this deployment to support a real mass evacuation, it is important to notice that each witness unit is able to provide simultaneous access to 30-40 people. This means that, through the simulated deployment, between 3000-4000 evacuees can simultaneously obtain information about their evacuation routes in few seconds. As mentioned, the evacuation routes information is lightweight, easy to merge and it is managed locally into the mobile device of the

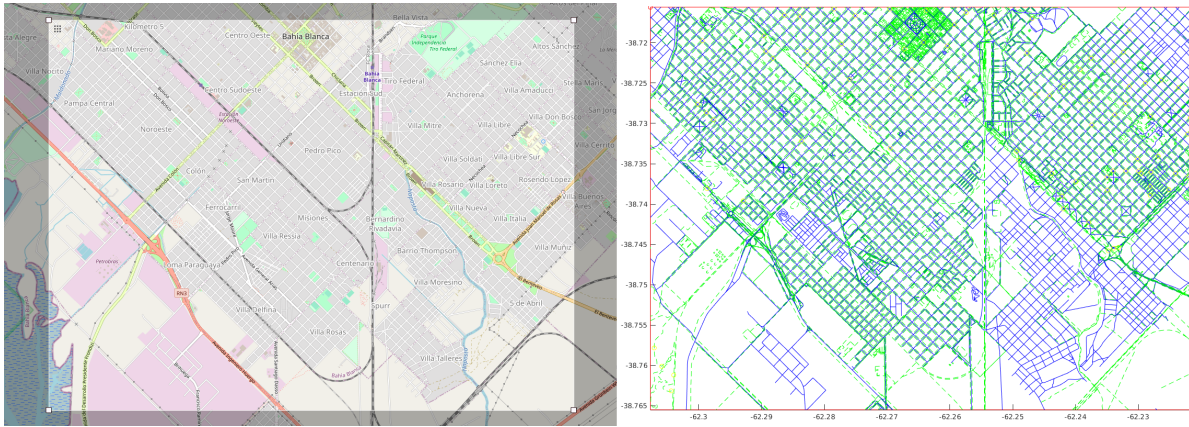


Fig. 9 Map of Bahia Blanca city

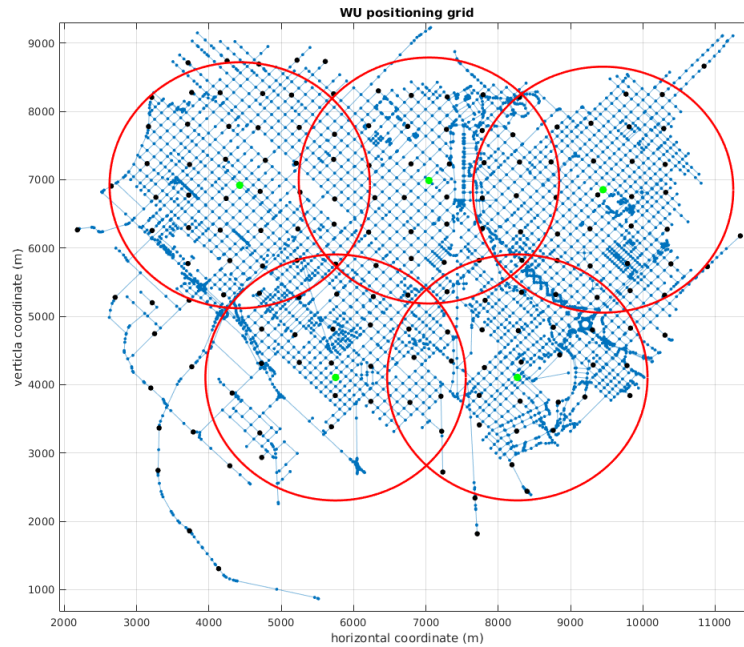


Fig. 10 Map for deploying Ws in Bahia Blanca city, considering $D = 500m$.

evacuees, therefore, the witness units can inform the evacuation routes to hundred of thousands of people in some few minutes. Next, we explain the simulation process and the obtained results.

6.2 System Simulation

For each witness unit, the number of events to be reported (i.e., messages to be sent to the shelter) was simulated with a random Poisson generator. This distribution was chosen because it models random events with a certain probability to occur in a certain time interval or area, by following an exponential distribution

that has the memoryless property. The rate of the Poisson distribution was varied between 1 and 10, which is equivalent to report between 1 and 50 events per witness unit.

The system was simulated 100 times for each possible rate, and the results were averaged. This represents a simulation scenario in which the network is stressed. More messages in the network implies more time waiting for being selected to be sent, until the increment in the traffic saturates the capacity of the communication network.

Figure 12 shows the latency in the access to the network. As it can be seen, the delay is proportional to the rate at which messages are generated in the witness

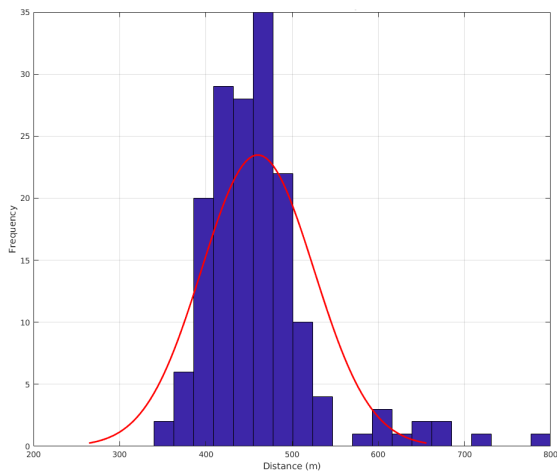


Fig. 11 Distribution of distances between neighbor nodes using $D = 500m$.

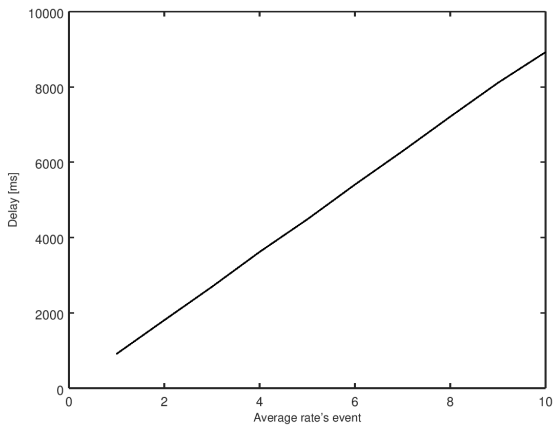


Fig. 12 Transmission delay in an infrastructure with 100 witness units

units. As the scheduling policy proposed is a simple round-robin, the average delay increments when more messages have to be sent.

Figure 13 illustrates the throughput of the network measured in $[\text{payload's bytes}]/[\text{s}]$. The absolute variation is not significant as a function of the the events generation ratio, less than 1%. This is because when the rate is low, the amount of information to send is also low; and when the rate is incremented, the delay in the network is also incremented, therefore, the final relation is similar. Finally, when the network is saturated, the throughput is independent of the number of events, as there is no more available bandwidth.

With this in mind, it is possible to predict the behavior of the system based on the amount of witness units required to support mass evacuations in a particular city or area. Clearly, there is a trade off between the

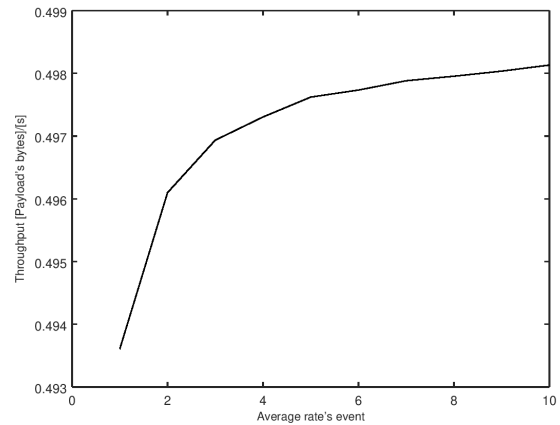


Fig. 13 Throughput for an infrastructure with 100 witness units

number of these units deployed and the final throughput of the network.

Several alternatives can be implemented to keep the throughput into reasonable levels, while still providing good breadcrumb trails for evacuees. This can be achieved by allowing some witness units to report events to the shelters, using dynamic priority policies to poll the witness units. Moreover, the system should allow simultaneous polling to witness units that are not within communication range, avoiding collisions. It can be done in cases like in Figure 10, where there are five shelters and approximately 170 witness units that may be paired to improve the response time. In this way the top left region can be scheduled simultaneously with the bottom right one, while the top right zone can be scheduled together with the bottom left. Finally, the central region is scheduled alone, as it has overlapped regions with all the other four. In this way, the average delay is reduced significantly and the throughput is incremented as two shelters receive updated information in a simultaneous way.

The capabilities shown by the simulated system is a consequence of using the proposed infrastructure, but also the set of design decisions that helps increase the limits and coverage of such an infrastructure. Particularly, the technology used to link the nodes, the density and location of witness units deployed in the field, the type of information exchanged between components, and the type of interaction allowed among nodes, are design aspects that must be addressed during the system design, since they have direct consequences on the system performance.

7 Conclusions and Future Work

This paper proposes an IoT-based infrastructure to provide support to civilians and first response organizations during large scale evacuation processes. This infrastructure covers both implicit and explicit interactions between the AmI system and the evacuees.

The formalization of this infrastructure was performed using an adaptation of IoT-Calculus, which allows modeling these systems and also evaluating their performance at preparedness time. Thus, the design of evacuation support systems can be adjusted based on the simulation results, and implemented only after they have shown to be suitable from a theoretical perspective.

A simple deployment algorithm was also proposed to determine suitable locations of Ws in an urban area, and thus provide ambient intelligence to evacuees while they are guided to shelters. The path to the shelters requires that the mobile units used by the evacuees connect to different Ws while people change from one location to another. These interactions among the system components are invisible for the end-users, who perceive the mobile application as an ubiquitous system that allows them access to information about their evacuation route anytime and anywhere. In this sense, the density of Ws, the technology used to connect the system components, and the type of information exchanged among these nodes play a key role, since the ubiquity perceived by the end-users and the limit of the system to collapse due to information overload depend on these aspects.

The proposed infrastructure implements a keep alive protocol that detects failures in the system components, and mechanisms to dynamically change the behavior of the nodes depending on the type and location of the fall to address. This behavior change also considers situations in which a node gets alive after a temporal collapse.

A proof-of-concept of the system was implemented using out-of-the-shelf components. Using data collected from a real scenario, a particular infrastructure was modeled to support mass evacuations in the urban area of the Bahia Blanca city, in Argentina. The capability of that system was evaluated using simulations, and the results show not only the feasibility of the proposal, but also its capability to deal with the evacuation process even when the network is overloaded.

Although the proposed infrastructure was designed particularly to support ambient intelligence in urban evacuation areas, it could be used with other purposes whether the systems consider particular implementation constraints; for instance, the deployment of Ws must ensure service coverage into the whole destina-

tion area, and the information exchanged by the nodes of the infrastructure should minimize the collapses due to information overload. In this sense, more experimental validation of the proposed infrastructure is required to determine its limits, not only to support large scale evacuation processes, but also to be applied in other application domains.

References

1. Abbasi, K.M., Khan, T.A., Haq, I.U.: Hierarchical modeling of complex internet of things systems using conceptual modeling approaches. *IEEE Access* **7**, 102772–102791 (2019)
2. Al-Nabhan, N., Al-Aboody, N., Islam, A.A.A.: A hybrid iot-based approach for emergency evacuation. *Computer Networks* **155**, 87 – 97 (2019). DOI <https://doi.org/10.1016/j.comnet.2019.03.015>
3. Araneda, J.C., Rudnick, H., Mocarquer, S., Miquel, P.: Lessons from the 2010 chilean earthquake and its impact on electricity supply. In: 2010 International Conference on Power System Technology, pp. 1–7 (2010)
4. Avvenuti, M., Cimino, M., Cresci, S., Marchetti, A., Tesconi, M.: A framework for detecting unfolding emergencies using humans as sensors. *Springerplus* **5**(43) (2016). DOI [10.1186/s40064-016-1674-y](https://doi.org/10.1186/s40064-016-1674-y)
5. Bi, H., Gelenbe, E.: A survey of algorithms and systems for evacuating people in confined spaces. *Electronics* **8**, 711 (2019). DOI [10.3390/electronics8060711](https://doi.org/10.3390/electronics8060711)
6. Conti, M., Passarella, A., Das, S.K.: The internet of people (iop): A new wave in pervasive mobile computing. *Pervasive and Mobile Computing* **41**, 1 – 27 (2017). DOI <https://doi.org/10.1016/j.pmcj.2017.07.009>
7. Diaz, P., Onorati, T., Romano, M., Aedo, I.: Designing affordable technologies to integrate citizens in early warning activities. *Proceedings* **2**, 1253 (2018). DOI [10.3390/proceedings2191253](https://doi.org/10.3390/proceedings2191253)
8. Eaton, J.W., Bateman, D., Hauberg, S., Wehbring, R.: GNU Octave version 4.2.1 manual: a high-level interactive language for numerical computations (2017). URL <https://www.gnu.org/software/octave/doc/v4.2.1/>
9. Eggly, G.M., Finochietto, J.M., Micheletto, M., Centelles, R.P., Santos, R., Ochoa, S.F., Meseguer, R., Orozco, J.: Evacuation supporting system based on iot components. In: Multidisciplinary Digital Publishing Institute Proceedings, vol. 31, p. 38 (2019)
10. Eismann, K., Posegga, O., Fischbach, K.: Decision making in emergency management: The role of social media. In: ECIS 2018 Proceedings, pp. 2–20 (2018)
11. Glushkova, T., Stoyanov, S.: Ambient-oriented modeling of intelligent context-aware systems. *Bulgarian Comput. Sci. Commun* **7**(1), 53–61 (2018)
12. Hossmann, T., Legendre, F., Carta, P., Gunningberg, P., Rohner, C.: Twitter in disaster mode: Opportunistic communication and distribution of sensor data in emergencies. In: Proceedings of the 3rd Extreme Conference on Communication: The Amazon Expedition, ExtremeCom '11, pp. 1–6. Association for Computing Machinery, New York, NY, USA (2011). DOI [10.1145/2414393.2414394](https://doi.org/10.1145/2414393.2414394)
13. Houston, J.B., Hawthorne, J., Perreault, M., Park, E., Hode, M., Halliwell, M., McGowen, S., Davis, R., Vaid, S., Mcelderry, J., Griffith, S.: Social media and disasters:

- A functional framework for social media use in disaster planning, response, and research. *Disasters* **39** (2014). DOI 10.1111/disa.12092
14. Huang, C.F., Tseng, Y.C.: The coverage problem in a wireless sensor network. *Mob. Netw. Appl.* **10**(4), 519–528 (2005). DOI 10.1007/s11036-005-1564-y. URL <https://doi.org/10.1007/s11036-005-1564-y>
 15. Kishorbhai, V.Y., Vasantbhai, N.N.: Aon: A survey on emergency communication systems during a catastrophic disaster. *Procedia Computer Science* **115**, 838 – 845 (2017). DOI <https://doi.org/10.1016/j.procs.2017.09.166>. 7th International Conference on Advances in Computing & Communications, ICACC-2017, 22-24 August 2017, Cochin, India
 16. Koussaifi, M., Trouilhet, S., Arcangeli, J.P., Bruel, J.M.: Ambient intelligence users in the loop: Towards a model-driven approach. In: *Federation of International Conferences on Software Technologies: Applications and Foundations*, pp. 558–572. Springer (2018)
 17. Lanese, I., Bedogni, L., Di Felice, M.: Internet of things: A process calculus approach. In: *Proceedings of the 28th Annual ACM Symposium on Applied Computing, SAC '13*, p. 1339–1346. Association for Computing Machinery, New York, NY, USA (2013). DOI 10.1145/2480362.2480615. URL <https://doi.org/10.1145/2480362.2480615>
 18. Lujak, M., Billhardt, H., Dunkel, J., Fernández, A., Hermoso, R., Ossowski, S.: A distributed architecture for real-time evacuation guidance in large smart buildings. *Computer Science and Information Systems* **14**, 257–282 (2017). DOI 10.2298/CSIS161014002L
 19. Malizia, A., Onorati, T., Diaz, P., Aedo, I., Astorga-Paliza, F.: Sema4a: An ontology for emergency notification systems accessibility. *Expert Systems with Applications* **37**(4), 3380 – 3391 (2010). DOI <https://doi.org/10.1016/j.eswa.2009.10.010>
 20. Miaoudakis, A.I., Petroulakis, N.E., Kastanis, D., Askoxylakis, I.G.: Communications in emergency and crisis situations. In: *Distributed, Ambient, and Pervasive Interactions (DAPI)*, International Conference on, pp. 555–565. Springer (2014). DOI 10.1007/978-3-319-07788-8_51
 21. Micheletto, M., Petrucci, V., Santos, R., Orozco, J., Mosse, D., Ochoa, S.F., Meseguer, R.: Flying real-time network to coordinate disaster relief activities in urban areas. *Sensors* **18**(5) (2018). DOI 10.3390/s18051662
 22. Mileti, D.: *Disasters by design: A reassessment of natural hazards in the United States*. Joseph Henry Press (1999)
 23. Milner, R., Parrow, J., Walker, D.: A calculus of mobile processes, part 1. University of Edinburgh Laboratory for Foundations of Computer Science (1989)
 24. Mulero Chaves, J., De Cola, T.: 1 - public warning applications: Requirements and examples. In: D. Câmara, N. Nikaiein (eds.) *Wireless Public Safety Networks* 3, pp. 1 – 18. Elsevier (2017). DOI <https://doi.org/10.1016/B978-1-78548-053-9.50001-9>
 25. Murayama, Y., Yamamoto, K.: Research on disaster communications. In: *IFIP Advances in Information and Communication Technology*, pp. 1–11. vol 516. Springer, Cham (2019)
 26. Nations, U.: *World population prospects 2019* (2019)
 27. Ochoa, S.F., Santos, R.: Human-centric wireless sensor networks to improve information availability during urban search and rescue activities. *Information Fusion* **22**, 71 – 84 (2015). DOI <https://doi.org/10.1016/j.inffus.2013.05.009>
 28. Onorati, T., Malizia, A., Diaz, P., Aedo, I.: Modeling an ontology on accessible evacuation routes for emergencies. *Expert Systems with Applications* **41**(16), 7124 – 7134 (2014). DOI <https://doi.org/10.1016/j.eswa.2014.05.039>
 29. Paul, P.S., Hazra, K., Saha, S., Nandi, S., Chakraborty, S., Das, S.: Generating crisis maps for large-scale disasters: Issues and challenges. this publication is an outcome of the r&d work undertaken in the itra project of media lab asia entitled. “post-disaster situation analysis and resource management using delay-tolerant peer-to-peer wireless networks (disarm)”. In: D. Câmara, N. Nikaiein (eds.) *Wireless Public Safety Networks* 3, pp. 67 – 98. Elsevier (2017). DOI <https://doi.org/10.1016/B978-1-78548-053-9.50004-4>
 30. Pueyo Centelles, R., Meseguer, R., Freitag, F., Navarro, L., Ochoa, S.F., Santos, R.M.: Loramoto: A communication system to provide safety awareness among civilians after an earthquake. *Future Generation Computer Systems* **115**, 150 – 170 (2021). DOI <https://doi.org/10.1016/j.future.2020.07.040>
 31. Párraga Niebla, C., Mulero Chaves, J., De Cola, T.: 8 - design aspects in multi-channel public warning systems. In: D. Câmara, N. Nikaiein (eds.) *Wireless Public Safety Networks* 2, pp. 227 – 261. Elsevier (2016). DOI <https://doi.org/10.1016/B978-1-78548-052-2.50008-6>
 32. Reina, D.G., Askalani, M., Toral, S.L., Barrero, F., Asimakopoulou, E., Bessis, N.: A survey on multihop ad hoc networks for disaster response scenarios. *International Journal of Distributed Sensor Networks* **2015**, 3 (2015). DOI 10.1155/2015/647037
 33. Reuter, C.: Communication between power blackout and mobile network overload. *International Journal of Information Systems for Crisis Response and Management (IJISCRAM)* (accepted) **6** (2014). DOI 10.4018/ijisrcam.2014040103
 34. Romano, M., Onorati, T., Aedo, I., Díaz, P.: Designing mobile applications for emergency response: Citizens acting as human sensors. *Sensors (Basel, Switzerland)* **16** (2016)
 35. Santos, R., Mosse, D., Znati, T., Comfort, L.: Design and implementation of a witness unit for opportunistic routing in tsunami alert scenarios. *Safety Science* **90**, 75 – 83 (2016). DOI <https://doi.org/10.1016/j.ssci.2015.09.014>. Building Community Resilience to Global Hazards: A Sociotechnical Approach
 36. Spialek, M.L., Houston, J.B.: The influence of citizen disaster communication on perceptions of neighborhood belonging and community resilience. *Journal of Applied Communication Research* **47**(1), 1–23 (2019). DOI 10.1080/00909882.2018.1544718
 37. Stieglitz, S., Bunker, D., Mirbabaie, M., Ehnis, C.: Sensemaking in social media during extreme events. *Journal of Contingencies and Crisis Management* **26** (2017). DOI 10.1111/1468-5973.12193