Pervasive and Mobile Computing 106 (2025) 101997

Contents lists available at ScienceDirect

Pervasive and Mobile Computing

journal homepage: www.elsevier.com/locate/pmc

Collective victim counting in post-disaster response: A distributed, power-efficient algorithm via BLE spontaneous networks

Giacomo Longo [a], Alessandro Cantelli-Forti [b], Enrico Russo [a,*], Francesco Lupia [c], Martin Strohmeier [d], Andrea Pugliese [c]

[a] DIBRIS, University of Genoa, Genoa, Italy
[b] RaSS National Laboratory, CNIT, Pisa, Italy
[c] DIMES, University of Calabria, Rende, Italy
[d] Cyber-Defence Campus, armasuisse S + T, Thun, Switzerland

ARTICLE INFO

Keywords:
Mobile devices
Bluetooth
Disaster response
Spontaneous wireless communications
Probabilistic data structure

ABSTRACT

Accurately determining the number of people affected by emergencies is essential for deploying effective response measures during disasters. Traditional solutions like cellular and Wi-Fi networks are often rendered ineffective during such emergencies due to widespread infrastructure damage or non-functional connectivity, prompting the exploration of more resilient methods. This paper proposes a novel solution utilizing Bluetooth Low Energy (BLE) technology and decentralized networks composed entirely of mobile and wearable devices to count individuals autonomously without reliance on external communication equipment or specialized personnel. This count leverages uncoordinated relayed communication among devices within these networks, enabling us to extend our counting capabilities well beyond the direct range of rescuers. A formally evaluated, experimentally validated, and privacy-preserving counting algorithm that demonstrates rapid convergence and high accuracy even in large-scale scenarios is employed.

1. Introduction

The era we live in is characterized by an unprecedented frequency of natural disasters, a growing fear of terrorist attacks, rising geopolitical tensions, and ongoing conflicts [1,2]. Given the often unpredictable nature of these events, effective emergency management capabilities are crucial. To this end, many frameworks have emerged to prepare where possible and respond to such emergencies [3]. These include the Office for the Coordination of Humanitarian Affairs (OCHA) guidance frameworks [4], the Sendai Framework [5], and the International Federation of Red Cross and Red Crescent Societies (IFRC) emergency response protocols [6].

A focal point concerns accurately and precisely counting individuals affected by disasters. Notably, the OCHA frameworks emphasize the efficient allocation of resources, which heavily depends on knowing the number of persons involved to prioritize aid and ensure it reaches the most people possible. Similarly, the Sendai Framework and IFRC protocols are designed to scale the response according to the size of the emergency and correlate the response effort with the number of people affected.

Wireless communication technologies have long been established as a practical solution for supporting delicate phases of emergency response [7,8], including counting those impacted. Their success continues to rise, driven by a fundamental aspect of contemporary society: the widespread ownership of mobile and wearable devices, which can effectively leverage these communication technologies.

* Corresponding author.
E-mail address: enrico.russo@unige.it (E. Russo).

https://doi.org/10.1016/j.pmcj.2024.101997
Received 17 July 2024; Received in revised form 17 October 2024; Accepted 18 November 2024
Available online 28 November 2024

In emergencies, the pervasive presence of these devices enables both *active* and *passive* measures. Active measures include owners sending and receiving alerts [9]. In contrast, passive measures monitor individuals who carry a device through existing wireless communications (e.g., registrations to the cellular network or Wi-Fi hotspots), preventing the necessity for direct user involvement.

Relying on active measures to count individuals can be impractical and unreliable. It requires users to take specific actions, which may not be feasible in chaotic or life-threatening situations. Passive measures are preferable, allowing responders to manage the counting process directly and ensure timely and more accurate results. A significant limitation of passive counting is that catastrophic events frequently make the supporting connectivity infrastructures unavailable due to electrical blackouts and disruptions. Possible actions consist of deploying temporary connectivity solutions, such as portable base stations [10] or International Mobile Subscriber Identity catchers [11]. However, these solutions require special equipment and personnel, can take time to deploy, and, most importantly, are effective within a limited radius from their placement [12].

Recently, leading device manufacturers have brought Offline Finding Networks (OFNs) [13,14] to the forefront. OFNs leverage direct and Bluetooth Low Energy (BLE) communications between devices, i.e., *spontaneous* networks, enabling crowdsourced searches for missing items. In particular, these networks offer a practical, low-energy solution for data exchange between devices without relying on external and vulnerable infrastructure. Consequently, BLE and the concept underlying OFNs have started playing a role in emergency management. As a prime example, they helped combat the COVID-19 pandemic by enabling contact tracing [15]. Contact tracing collects and stores the owner's contacts locally through BLE advertisements from nearby devices. If an individual tests positive for COVID-19, locally stored data help automate notifying those who have been in close contact with the infected individual.

This work builds upon the advantages of BLE technology and spontaneous networks to propose an innovative solution for the passive counting of individuals during emergencies. After activation, the presented solution enables devices involved in an emergency to collaborate and passively count victims using a probabilistic data structure that contains no personally identifiable information (PII). First responders can integrate their mobile devices into this spontaneous and collaborative network to quickly assess the situation. Notably, this solution does not require specialized equipment and avoids the limitation of counting only BLE advertisements received within a specified time period. This approach overcomes the detection constraint to devices in direct line of sight, as is the case with contact tracing and OFNs.

The strengths of the presented proposal are outlined below.

- It relies on devices already available to first responders and affected individuals, eliminating the need for additional equipment or highly specialized personnel in the field.
- It can count individuals even beyond the BLE range of the rescuers by relying on spontaneous network bridges formed between the smartphones of other victims. This *store-and-forward* functionality can be added to existing devices through an automated software upgrade, as demonstrated by the contact tracing APIs introduced by Google and Apple under European Union guidelines [16].
- It employs a counting algorithm that has been (i) formally evaluated and proven to be convergent and (ii) experimentally proven to provide high accuracy with quick convergence, even when dealing with large-scale emergency scenarios involving numerous individuals and devices.
- In light of privacy concerns regarding contact tracing systems [17,18], the proposed solution ensures that it does not expose, exchange, or store remotely any data about the device owner. Indeed, the counting process uses a random identifier, which cannot be traced back to the user.

Overall, the main contributions of the paper can be summarized as follows.

- The proposal of an innovative solution for passive counting of individuals in emergencies using standard mobile and wearable devices and their capability to create BLE-based spontaneous networks. In particular, independence from external infrastructure or equipment and the coverage of large areas allow first responders to quickly and effectively organize efforts, as major international emergency response frameworks recommend.
- The conduction of a comprehensive measurement campaign in a real and complex environment, already affected by a severe and widely remembered emergency in the past, to study the behavior of BLE and devices.
- *An extensive assessment* of the proposed solution using the measures gathered from the previous campaign and simulating representative scenarios with hundreds of collaborating devices.

Structure of the paper. The remainder of the paper is organized as follows. In Section 2, the related work is revised. Section 3 presents background concepts relevant to this work. Section 4 details the proposed methodology. Section 5 describes the case study and illustrates the experimental activity to validate the proposed methodology. Section 6 discusses the performance and viability of the proposed system, addresses security and privacy considerations, and examines threats to validity. Finally, Section 7 draws the conclusions.

2. Related work

Various approaches have been proposed for counting and locating victims in disaster scenarios, leveraging different wireless technologies and network architectures [19–22]. This section categorizes existing methods based on their underlying technologies and discusses their applicability in emergency situations.

(a) Wireless-based Emergency Networks. Several studies have explored the use of wireless technologies such as BLE for establishing communication networks during disasters. For example, G. Chatzimilioudis et al. [23] developed algorithms that operate in emergency situations where cellular networks are down. These algorithms create a network using smartphones' short-range communication technology, such as BLE, enabling users to interact with their closest neighbors. However, the proposed solution exploits centralized algorithms, which makes them unsuitable in a natural disaster scenario where infrastructure may be compromised or nonexistent.

An approach relying on a distributed scheme is presented in [24], where spontaneous emergency networks are created out of mobile end-user devices by allowing each device to autonomously determine its role as either a router or a gateway, based on its available energy. Routing paths are then established using swarm intelligence techniques, facilitating the formation of a dynamic network. Despite its advantages, this approach suffers from slow convergence and a high volume of message exchanges, making it impractical for BLE-enabled devices with limited energy. Another similar solution is proposed in [25] which also introduces opportunistic networking, mobile sensing, and distributed information processing for emergency management. Regarding this study, we highlight that only people with the right software can participate in the creation of the network, which limits the practical applicability in real scenarios.

The Insight system discussed in [26] demonstrates an alerting mechanism that detects signals from BLE beacons indicating danger zones without relying on the internet or communication infrastructure. However, compared to our approach, the authors do not address security and privacy issues.

Previous research has also investigated the use of Wi-Fi technology for emergency scenarios. For instance, in [27], the authors suggest employing the Wi-Fi Direct module to establish an opportunistic rescue network, though their approach supports only single-hop communication. In contrast, the approach presented in [28] introduces a method aimed at efficient multi-hop information dissemination within emergency networks. Their approach organizes devices into virtual cells that can dynamically form or disband to enable packet exchange. Nevertheless, this method relies on the assumption that all devices are time-synchronized, which may limit its applicability.

The Smartphone-Assisted Victim Localization (SmartVL) method, detailed in [29], leverages smartphones to locate victims in disasters. This method involves smartphones monitoring changes in the radio environment to automatically switch to a disaster mode. Once in disaster mode, the smartphones transmit periodic emergency messages containing their location coordinates to nearby devices carried by first responders and rescuers. Contrary to the presented proposal, which relays information also to nodes situated further away, those messages are accessible just when in direct proximity of the victim.

Other studies [30,31] have exploited signal processing techniques for proximity and distance estimation. Unlike these works, the proposed approach does not rely on radio characteristics, such as the Received Signal Strength Indicator (RSSI) or the setup of point-to-point links. RSSI is intentionally excluded due to its susceptibility to variations in transceiver power and receiver sensitivity, as well as environmental factors impacting radio propagation. As per point-to-point links, these are avoided because they require constant communication, leading to high energy consumption that could prematurely drain the victim's battery and shorten the window in which the emergency system is operational.

(b) People Counting and Crowd Sensing. Methods focusing on people counting and crowd sensing have been proposed using BLE technology. For example, an approach for people counting based on passive presence detection was proposed by Brockmann et al. [32].

However, this method focuses on counting people in waiting lines rather than those affected by emergencies. Their system requires the setup of additional BLE transceivers and relies on a fixed installation, which may be impractical during natural disasters. The proposed approach utilizes spontaneous networks formed by existing mobile devices, making it suitable for chaotic environments with compromised infrastructure. In [33], the authors present a method for sensing people by analyzing crowd dynamics and movement patterns using a network of BLE tags carried by individuals. This method does not focus on counting and raises potential privacy concerns due to the tracking of individual movement patterns.

(c) Infrastructure-Dependent Solutions. As per emergency systems not based on BLE technology, temporary and portable mobile base stations (OpenBTS) have been explored for locating individuals, as discussed in [34–36]. However, these solutions depend on specialized equipment that may delay rescue operations. The large coverage associated with cellular base stations could also lead to a "false alarm" scenario, where devices far away and not in need of help are erroneously included within the emergency area. Furthermore, tracking cell phones by SIM identifiers requires coordination with mobile network operators to shut down their towers and poses significant privacy concerns.

3. Background

This section provides a comprehensive overview of the foundational concepts and technical aspects relevant to the presented methodology.

3.1. Bloom filters

Bloom filters [37] are highly efficient probabilistic data structures that allow for approximate membership queries, i.e., determining if an element might be present in a set or is definitely not present.[1] Each Bloom filter $B = \langle b_{1\ldots m}, h_{1\ldots k} \rangle$ is comprised of a bit vector b of size m, and k different hash functions h.

Adding an element e to a Bloom filter involves applying each of the k hash functions to that element, which generates k distinct indices in $[0, m)$. For each generated index, the corresponding index in b is then set to 1, as shown in Eq. (1).

$$B \bigcup \{e\} = \left\langle b \bigvee \left[\bigvee_{i=1}^{k} h_i(e) \right], h \right\rangle \tag{1}$$

One advantageous property of Bloom filters is their ability to be merged together using a simple and idempotent operation, the logical OR of their underlying bit vector, provided that both filters have the same length and underlying hash functions (Eq. (2)).

$$B_1 \bigcup B_2 = B_1\langle b_1, h_1 \rangle \bigcup B_2\langle b_2, h_2 \rangle = \langle b_1 \bigvee b_2, h \rangle \qquad \text{if } m_1 = m_2, h_1 = h_2 = h \tag{2}$$

An estimation of the number of elements $\hat{n}$ in a Bloom filter can be calculated [38] using Eq. (3).

$$\text{let } X = \text{number of bits set to one} \qquad \hat{n} = \begin{cases} 0 & \text{when } X = 0 \\ \frac{m}{k} & \text{when } X = m \\ -\frac{m}{k} \ln\left(1 - \frac{X}{m}\right) & otherwise \end{cases} \tag{3}$$

Instead, Eq. (4) allows to calculate the optimal number [39] of hash functions $\hat{k}$ for a Bloom filter containing N elements.

$$\text{let } N = \text{estimated number of elements} \qquad \hat{k} = \frac{m}{N} \ln(2) \tag{4}$$

3.2. Bluetooth low energy

Bluetooth Low Energy (BLE) is a wireless communication technology designed for short-range and low-power applications. Similar to Bluetooth Classic, BLE operates in the same unlicensed 2.4 GHz ISM band but leveraging a distinct, more efficient physical layer. Thanks to this improved efficiency, it has found an almost ubiquitous presence among power-constrained devices like mobile phones or in the Internet of Things.

The main capability offered by BLE is advertisements. BLE advertisements are small packets of data that are broadcast by BLE devices to announce their presence and capabilities to surrounding devices. These packets can contain up to 31 bytes of data and serve various purposes, including device discovery, connection establishment, and transmission of small data payloads. Advertisements allow devices to broadcast and receive data without establishing a continuous connection.

One such advertisement, known as service data advertisement, allows one to associate a 128-bit service UUID with an arbitrary payload. This capability enables devices to broadcast information to devices interested in the given service UUID. As per the size of such data, each advertisement packet includes a 2-byte preamble, a 4-byte device address, and a 4-byte for the service UUID, corresponding to a minimum supported payload size of 13 bytes for any device compliant with the Bluetooth specification, which mandates a minimal transmission unit (ATT_MTU) of 23 bytes [40, 5.2.1].

This figure of 13 bytes is only a conservative estimate, as some BLE devices support extended advertisements, which significantly increase the maximum data payload from 31 bytes to up to 1650 bytes, but this feature requires compatible hardware on both the sending and receiving ends.

Transmission-wise, in both Android and iOS, BLE advertisements can be configured to be sent in three different modalities: low power, medium, and low latency. Each modality corresponds to an increased duty cycle, with higher modes transmitting more advertisements per unit of time. Details of such duty cycles are device-dependent. Section 5.3.1 includes example values for one such device. While higher duty cycles enhance the likelihood of a given advertisement being received by other devices, it also results in higher power consumption.

Reception of BLE advertisements is very power-efficient. BLE chips are equipped with hardware filters that allow them to process incoming advertisements and wake the device CPU only when an advertisement matches specific criteria. For service data advertisements, these filters can match advertisements for a given UUID and a mask on its associated payload, ensuring that the CPU remains in a low-power state unless relevant data is received.

Another power-saving feature is advertising sets, which allow multiple advertisements to be preloaded and cycled through by the radio hardware itself. This feature reduces the need for the CPU to frequently update advertisement contents, thus conserving energy and improving efficiency.

BLE advertisements can reach distances of up to hundreds of meters, depending on environmental conditions and device configurations [41]. In contrast, BLE point-to-point links typically have a range of up to ten meters and come with increased power consumption compared to advertisements.

[1] Bloom filters are initialized with a fixed bit vector size m and a fixed number of hash functions k, which remain constant throughout their use. The capacity of the Bloom filter to represent different elements (e.g., device identifiers) without significant false positives depends on these parameters and the number of elements added.

4. Methodology

This section describes the proposed methodology. After outlining design goals, assumptions, and constraints, the mechanisms that trigger the system into its active state are described. Then, an algorithm for counting people in the event of a disaster is discussed, formalized, and developed. Throughout this paper, devices are assumed to be naturally organized into a connected interaction network. Specifically, reference is made to a single interaction network, although there may be multiple such networks operating independently in practical scenarios.

4.1. Design goals and assumptions

The principal aim of the proposed system is to estimate the number of individuals within an emergency area. As a result, first responders can promptly receive this information, enabling the system to aid them in their initial planning and resource allocation.

In achieving this objective, the following design goals are essential to ensure the effectiveness and reliability of the system.

- It must propagate such estimates in a decentralized manner across all participating devices, allowing access to this information through contact with any node in the network.
- It must be *independent*, i.e., operating without relying on external infrastructure like cellular networks or Wi-Fi, which may be unavailable during emergencies, except for devices typically carried by individuals.
- It has to be *efficient*, i.e., minimize its power consumption in both its *idle* and *active* states and enable devices to either participate in the network for longer or survive long enough for other recovery methods to become available.

Further addressing efficiency, the system is *idle efficient* if its presence in a deactivated state has a negligible impact on the overall power consumption of devices. Similarly, the system is *active efficient* by minimizing power consumption when it runs and leveraging the available hardware and software capabilities to the fullest.

Moreover, the proposed system operates under the following assumption.

- A software component implementing the algorithm used by the system (see Section 4.3) is installed on devices.
- The individuals' devices are equipped with BLE radios compliant with Bluetooth Core Specifications [40], enabling the transmission and reception of advertisements.
- Each device is equipped with the most basic BLE radio module the standard allows, i.e., one capable of transmitting at most 13 bytes in a service data advertisement.

Regarding the first assumption, the software component must run as a service in Android or a background task in iOS. It is worth noting that the unattended installation of such a component requires an agreement with the platform providers. However, this is not unprecedented. Similar provisions were implemented during the COVID-19 pandemic for contact tracing applications, where both Google and Apple collaborated with public health authorities to integrate a service directly into their mobile platforms. Given the critical nature of the emergency response, the proposed system, aimed at aiding people in distress, may align with the interests of local, national, or supranational authorities, potentially warranting a similar arrangement.

4.2. Triggering

In order to meet the power efficiency requirement, the system has to stay in a quiescent state until some condition suggests that an emergency is in progress, triggering its transition to the active state.

Given the absence of a universally optimal strategy, a heuristic method was chosen to detect the onset of an emergency. This choice eliminates the need for active intervention from users, who may be incapacitated or otherwise unable to respond during an emergency.

The system's active state is triggered by the perceived unavailability of conventional communication infrastructure. The sudden absence of network coverage was selected as the primary alert condition, strongly suggesting a disruption in essential infrastructure often associated with emergency scenarios. Moreover, as most mobile operating systems provide robust facilities for waking up service in response to changes in network activation, this strategy allows the system to meet its desired idle efficiency goal.

To summarize, the service operates in a quiescent state, continuously monitoring network connectivity. Upon detecting a prolonged interruption in network coverage, the system transitions into alert mode. This state persists until a stable and sustained re-establishment of network connectivity returns the system to its idle state.

This triggering heuristic is prone to false positives, as a sudden absence of connectivity does not always imply an immediate need for intervention, e.g., in railway tunnels. However, since activation of the network does not threaten user privacy, the system's likelihood to intervene aligns with traditional safety systems, which prioritize false positives over false negatives, justifying this design choice. Furthermore, during the large-scale deployment of contact tracing, it was allowed to run continuously. This continuous operation was considered advantageous despite a slight increase in power usage. Since the actions associated with contact tracing – BLE advertisement and listening – are similar to those of the presented proposal in its active state, those false activations could be tolerated according to a similar compromise between utility and (potential) power consumption.

Finally, manual activation of the service in its passive mode should only be allowed when it is necessary to access the current node count estimate from nearby nodes, as in the case of emergency personnel.

4.3. Algorithm

This section outlines the proposed algorithm, beginning with a high-level overview, followed by a theoretical explanation, and concluding with practical considerations for implementing the algorithm in a way that fully utilizes hardware capabilities.

4.3.1. High-level description

At a high level, the proposed system functions by constructing a spontaneous network that shares an ever-increasing counter among its nodes without any coordination or point-to-point communication between them. This counter is expected to eventually converge toward an approximation of the number of nodes belonging to the network — that is, the total count of individuals within the emergency area.

The network consists of endpoints that can take either active or passive roles. Active nodes aim for their presence to be acknowledged in the overall counter and are available for transmitting the current count value. Passive nodes, on the other hand, merely wish to gather the latest node count estimate without exerting any side effects on the network. In the use case, active nodes represent devices owned by disaster victims, while passive nodes correspond to emergency responders' equipment. Any node that becomes activated due to the condition specified in the previous section will commence its operation as an active node.

Upon entering its active mode, each node establishes an internal Bloom filter with a bit vector size of m and predefined hash functions denoted by h, which are common to all running nodes. Next, each node generates its own random identifier, adding it to its local Bloom filter instance. Following this step, the nodes with active roles begin broadcasting their individual Bloom filters using BLE advertisements indefinitely. Each node – irrespective of its role – listens for such broadcasts and integrates the contents of each received Bloom filter into its own by performing merge operations. After an adequate number of iterations following this procedure, every node will have accumulated a local Bloom filter populated with all unique node identifiers, allowing to estimate from it the count of nodes as shown in Eq. (3). This conclusion is formalized in Theorem 1, described below.

Fact 1 (*Union of Bloom filters*)**.** *The union of two Bloom filters B_i and B_j results in a Bloom filter B_{ij} representing the union of the sets represented by B_i and B_j, i.e. $B_{ij} = B_i \bigcup B_j$.*

Fact 2 (*Commutativity and Associativity*)**.** *The union operation on Bloom filters is commutative, i.e. $B_i \bigcup B_j = B_j \bigcup B_i$, and associative, i.e. $B_i \bigcup (B_j \bigcup B_k) = (B_i \bigcup B_j) \bigcup B_k$.*

Fact 3 (*Idempotency of the Union Operation*)**.** *The union of a Bloom filter with itself does not change its contents, i.e., $B_i \bigcup B_i = B_i$.*

Theorem 1. *Let N be the number of devices in a connected interaction network, where each device i has a unique identifier I_i. Suppose each device initializes a Bloom filter B_i with its own identifier I_i, and all Bloom filters are configured with a bit vector size m and k hash functions suitable for estimating up to N elements. Then, assuming reliable communication and eventual message propagation throughout the network, the Bloom filter B_i of every device i will converge to a Bloom filter B^* that represents the union of all device identifiers, i.e., $B^* = \bigcup_{i=1}^{N} \{I_i\}$.*

Proof. Theorem 1 will be proved by induction on the number of devices n.

Basis Step. Consider a network with a single device, d_1, having a unique identifier I_1. The Bloom filter B_1 for this device is initialized with I_1. Since there are no other devices, no union operations are necessary. Thus, the Bloom filter B_1 correctly represents the set I_1.

Inductive Hypothesis. Assume that in a network of t devices with $1 \leq t \leq N-1$, all devices have Bloom filters that have converged to represent the union $\bigcup_{i=1}^{t} \{I_i\}$.

Inductive Step. Let the new device joining the network be d_{t+1} with unique identifier I_{t+1}. Device d_{t+1} initializes its Bloom filter B_{t+1} with I_{t+1} and begins broadcasting B_{t+1}. Due to the connected component assumption of the network, each of the t existing devices receives the B_{t+1} and updates their Bloom filters $B_i' = B_i \cup B_{t+1}$, for $i = 1, 2, \ldots, t$. By Fact 1, B_i' now represents $\bigcup_{i=1}^{t+1} \{I_i\}$.

Similarly, device d_{t+1} receives Bloom filters from existing devices, which by the inductive hypothesis represent $\bigcup_{i=1}^{t} \{I_i\}$, and it updates its Bloom filter $B_{t+1}' = B_i \cup B_{t+1}$, for each received B_i.

By Facts 2 and 3, the repeated and unordered union operations will not affect the final result. Since the network is connected and messages are reliably propagated, it is guaranteed that after sufficient time, all devices will have converged Bloom filters. Thus, by induction, the theorem holds for N devices. □

4.3.2. Detailed description

Next, the main routine of the system is described in detail. The system configuration parameters are as follows:

- M determines the number of BLE advertisements over which a Bloom filter broadcast will be spread. Since the size of such broadcasts is fixed at 13 bytes, this parameter governs the Bloom filter bit vector size. Higher values for M will increase the Bloom filter's maximum capacity and overall accuracy but require more advertisements for it to be received by other nodes.
- N represents an estimation of the number of nodes that need tracking by the system.
- ACT indicates whether a node takes on an active role (1) or a passive one (0).

- $BURST\_START$ is an optional optimization that enables the algorithm to converge faster by sending more advertisements during its initial phase. It is parameterized by two sub-parameters: T_{HM}, which indicates when the advertising frequency will be reduced, and T_{ML}, which specifies when the advertising frequency will be reduced to the most power-efficient value.
- T_{dt} specifies the allowed emergency elapsed time skew between nodes. It helps reduce the likelihood of the system confusing devices that entered the emergency state at significantly different times.

Algorithm 1 Listening/counting routine

Input: $16 \geq M > 0, N > 0, T_{ML} > T_{HM} > 0, T_{dt} > 0, ACT \in \{0,1\}, BURST\_START \in \{0,1\}$

```
1:  bloom ← MAKE_BLOOM(12 · 8 · M, (12·8·M)/N · ln(2))          ▷ Initialize Bloom filter
2:  loop
3:      BLOOM.RESET
4:      if ACT then
5:          WAITALERT
6:          id ← MAKEID                                          ▷ Generate a unique node ID
7:          BLOOM.ADD(id)                                        ▷ Active nodes initial broadcast is their ID
8:          SPAWN(BLEADVERTISEMENTLOOP)
9:      end if
10:     t_0 ← NOW                                                ▷ Record the start time
11:     bloom_slots ← MAKEBLOOMSLOTS(M, N)
12:     while ISALERT or not ACT do
13:         adv ← NEXTADVERTISEMENT                              ▷ Receive advertisement
14:         t_e ← MINUTES(NOW - t_0) mod 7
15:         Δt ← min(|t_e − adv.epoch|, 7 − |t_e − adv.epoch|)   ▷ Calculate time since alert
16:         if Δt > T_dt then
17:             continue                                         ▷ Discard advertisement if time incompatible
18:         end if
19:         BLOOM_SLOTS.ADDSLICE(adv.device, adv.payload, adv.slice_i)
20:         if BLOOM_SLOTS.COMPLETE(adv.device) then
21:             BLOOM.LOCK
22:             BLOOM.MERGE(BLOOM_SLOTS.GET(adv.device))
23:             UPDATEESTIMATE(BLOOM.ESTIMATENUMELEMENTS())
24:             BLOOM.UNLOCK
25:             BLOOM_SLOTS.CLEAR(adv.device)
26:         end if
27:     end while
28:     if ACT then
29:         KILL(BLEADVERTISEMENTLOOP)
30:     end if
31: end loop
```

Algorithm 1 implements the main logic to be performed by both active and passive nodes. The algorithm starts by initializing a Bloom filter with a size of $12 \cdot 8 \cdot M$ bits and the optimal number of hash functions to use, following the formula in Eq. (4) (Line 1). The filter uses 12 bytes instead of the available 13 because a byte at the beginning of the advertised data is used for additional information, which will be detailed below. Then, an endless loop is started. At the beginning of it, the bloom filter contents are cleared (Line 3).

When a node has an active role, the service is halted until awakened by the triggering of the alert condition described in Section 4.2 (Line 5). Post-trigger, the (active) node generates a unique id (Line 6) and commences running Algorithm 2 in another thread (Line 8).

Regardless of their role, nodes store the starting time in a variable t_0 (Line 10) and create a “Bloom slot" data structure (Line 11). This structure maps at most N BLE hardware addresses to the M slices of the Bloom filter each node transmits. It also tracks which of these slices has been populated.

The system then continuously listens for BLE service data advertisements (Line 13). Active nodes listen until the emergency ends, while passive ones repeat the procedure forever (Line 12). Upon receipt, each advertisement emergency start time is compared with the threshold value T_{dt} to prevent merging nodes that entered the emergency state at significantly different times (Line 16). This calculation occurs modulo 7 (Line 14), limiting the amount of dedicated metadata space in the advertised packet to 3 bits.

Once the time skew check passes, the Bloom slots data structure updates with the newly received Bloom filter slice (Line 19). Upon such an update, the system checks if a complete bloom filter can be assembled from the current device advertisements (Line 20). If possible, the local Bloom filter is atomically merged according to Eq. (2) with the content of the remote Bloom filter found in the structure (Line 22).

After merging, the estimated number of nodes updates using Eq. (3) (Line 23), and the “Bloom slot" data structure entry for the device is cleared (Line 25). Lastly, if an active node records the end of the emergency, it halts the advertisement routine (Line 29).

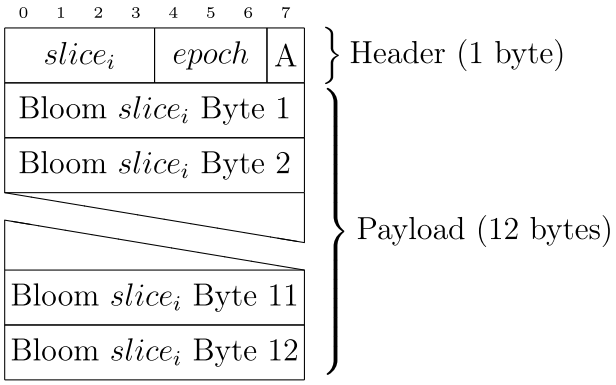


Fig. 1. Advertisement Payload Structure (13 bytes).

Algorithm 2 Advertisement routine

Input: $bloom, M > 0, BURST\_START \in \{0,1\}, T_{ML} > T_{HM} > 0, t_0$

1: $mode \leftarrow \begin{cases} HIGH & \text{if } BURST\_START \\ LOW & \text{otherwise} \end{cases}$ ▷ Set initial advertisement mode
2: **loop**
3: BLOOM.LOCK
4: $bloom\_snapshot \leftarrow$ BLOOM.COPY ▷ Atomically snapshot the bloom filter
5: BLOOM.UNLOCK
6: $t_a \leftarrow$ MINUTES(NOW - t_0) **mod** 7 ▷ Calculate time since alert status
7: **for** $slice_i \in 1$ **to** M **do**
8: ADVERTISE($mode$, $slice_i$, t_a, 1, BLOOM.SLICE($slice_i$))
9: **end for**
10: **if** $BURST\_START$ **then**
11: **if** $mode = HIGH$ **and** SECONDS(NOW $- t_0$) $> T_{HM}$ **then**
12: $mode \leftarrow MEDIUM$ ▷ Set advertisement to medium duty cycle
13: **else if** $mode = MEDIUM$ **and** SECONDS(NOW $- t_0$) $> T_{ML}$ **then**
14: $mode \leftarrow LOW$ ▷ Set advertisement to low duty cycle
15: **end if**
16: **end if**
17: **end loop**

Algorithm 2 implements the advertisement background routine. It begins by selecting an initial advertiser duty cycle: the lowest possible (LOW) if the BURST_START optimization is omitted, or the highest otherwise (Line 1). Then, an endless loop commences.

Inside the loop, the current Bloom filter contents are atomically copied (Line 4), and the time since the emergency start at which the Bloom filter was copied is saved in t_a (Line 6). Subsequently, each Bloom filter slice is sent as a separate advertisement (Line 8).

Fig. 1 illustrates the contents of the sent service data packet: it comprises 1 byte of metadata and 12 bytes of payload, totaling 13 bytes. Within the metadata byte, 4 bits (0-3) specify the index of the currently transmitted slice, 3 bits (4-6) indicate the elapsed time in minutes since the emergency onset, and a final bit, always set to one, is reserved for potential future use. The payload consists of 12 bytes extracted from the $slice_i$th division of the Bloom filter bit vector.

Finally, if the BURST_START optimization is enabled, the elapsed time since the start of the emergency in seconds is compared with T_{HM} to denote when the duty cycle switches to its medium setting (MEDIUM) and T_{ML} for when it is reset to its minimum value (Line 10).

4.4. Practical considerations

The implementation of the algorithms shown in the previous section needs some further changes in order to maximize the utilization of hardware resources and meet the “active efficient" requirement. First of all, in Algorithm 1, the check shown in Lines 14–16 has to be replaced with a dedicated filter on bits 4-6 of the received advertisement. This provision means that the device radio itself will take care of discarding time skewed advertisements without the involvement of the main processor. Then, in Algorithm 2, the for loop cycling through the various advertisements (Line 8) has to be replaced with the setup of an advertisement set containing one payload for each slice. In such a way, the radio itself will handle cycling through the various advertisements, requiring no intervention by the processor. By applying these changes, the system can fully leverage the hardware features offered by the

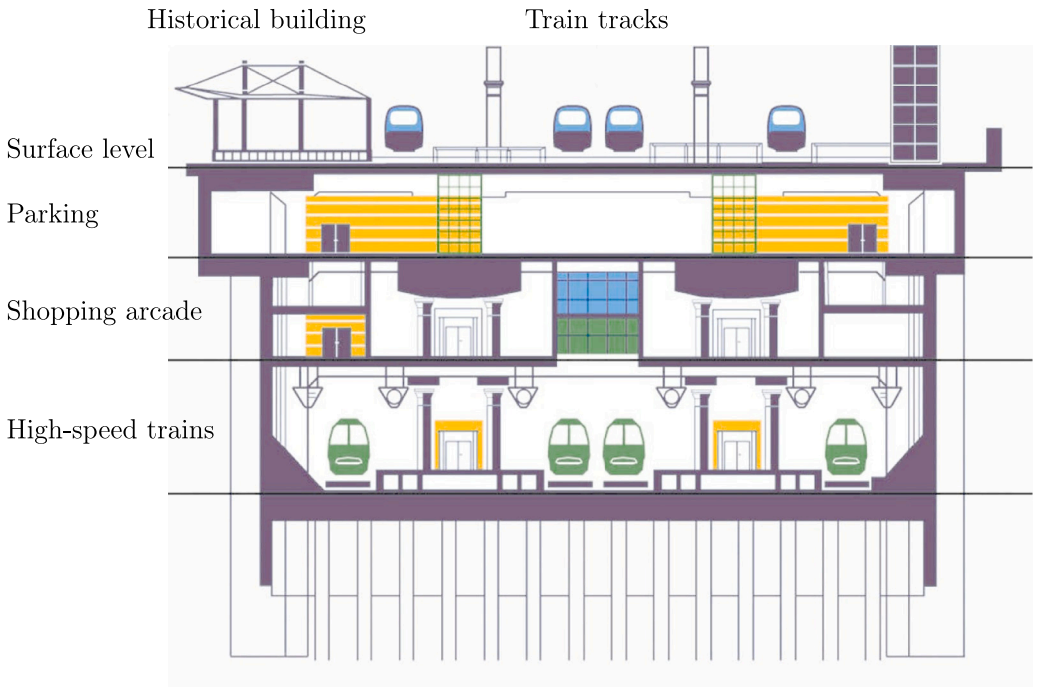


Fig. 2. Bologna Central Station's sectional elevation [44].

hardware and minimize its power usage by reducing the need to activate the power-hungry device CPU. Those optimizations closely match the recommendations given by Apple and Google for minimizing power usage while listening for BLE advertisements [42].

5. Experimental evaluation

5.1. Case study

The Bologna Central Station has been selected as the case study due to its historical significance and various complexities. On August 2, 1980, an Improvised Explosive Device (IED) detonated within the station, resulting in 291 injured persons (73 of whom died at the scene) [43] and lengthy, complicated rescue operations for survivors. The complexities include a high volume of passengers, an extensive surface area, and critical tracks located as much as 23 m below ground level, which potentially shield electromagnetic waves.

Briefly, this railway hub experiences a high volume of train movements, approximately 800 per day, and about 180,000 passengers. It operates as a through station, including a historic building and a newer wing for high-speed (AV) trains.

Fig. 2 provides its sectional elevation. The surface level includes the historic station building, train tracks, and two sets of side tracks dedicated to regional trains. In 2008, construction of a new three-level underground station began beneath the existing facility. This subterranean structure extends 642 m (2,106 ft 4 inches) in length and spans 56 m (183 ft 9 inches) in width, comprising three underground levels. The top underground level, situated at −7 m (−23 ft), is designated for parking, as well as pick-up and drop-off areas. The middle level, at −15 m (−49 ft 3 inches), houses a shopping arcade. The deepest level, at −23 m (−75 ft 6 inches), features four platforms specifically for high-speed trains.

5.2. Experimental settings

A two-phase experimental campaign was conducted. The initial phase was performed on-site at the Bologna Central Station in close collaboration with railway police. The objective was to collect data and coefficients necessary for the subsequent phase involving a discrete event simulator.

This process enabled us to accurately determine all relevant quantities involved, which in turn allowed us to perform a representative evaluation without necessitating a costly experiment involving hundreds of collaborating devices.

The first phase was performed with a bespoke application, depicted in Fig. 3, running on two Lenovo TB-X605XL tablets equipped with Android 9 and the BLE radio found within their Qualcomm Snapdragon 450 System On a Chip [45].

The application performs in parallel four tasks, as detailed above.

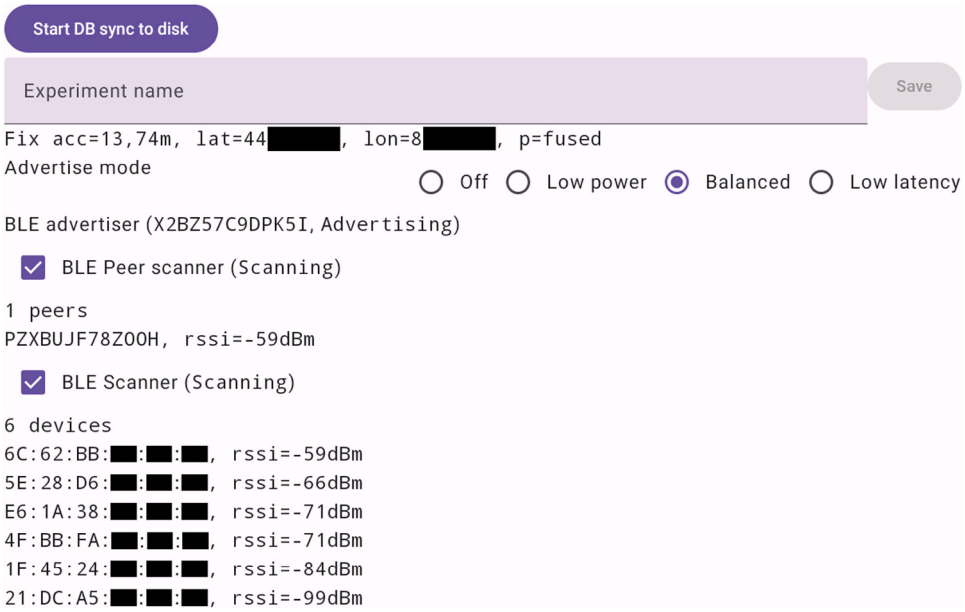
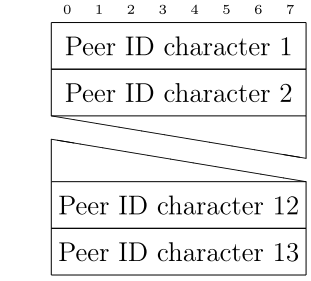
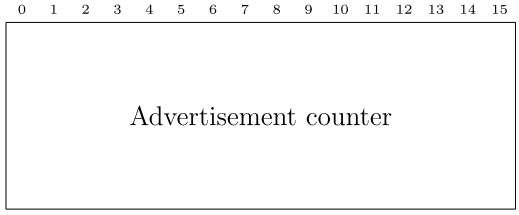


Fig. 3. Application used for the experiments.



(a) Peer ID broadcast payload (13 bytes).



(b) Advertisement metadata broadcast payload (8 bytes).

Fig. 4. Broadcast payloads used by the ranging experiments.

1. It records the current timestamp, position, and associated accuracy. The Google Play Services Fused Location API [46] is used, which merges GNSS, inertial, and network-derived measurements.
2. It runs, according to the selected duty cycle, an advertising set announcement containing either a generated peer id (Fig. 4(a)) or an ever-increasing 64-bit counter (Fig. 4(b)). Such values and associated timestamps are recorded in a local database.
3. It listens to the advertisements generated by the previous task coming from other devices, storing within its local database the peer ID, counter value, and measured Received Signal Strength Indicator (RSSI[2]).
4. It listens to any BLE advertisement, keeping track of the device address, RSSI, and timestamp of reception, saving it into a local database.

During the entire experimental campaign, which lasted for approximately 6 h, both devices had recorded:

- 13,124 position measurements.
- 68,721 BLE ranging advertisements coming from the developed application.
- 48,647 BLE advertisements, belonging to 3284 distinct devices.

Finally, security personnel from the station periodically reported the current occupancy of the instrumented area, allowing us to carry out the measurements from 5.3.2.

For the second phase, a discrete event simulator executing Algorithms 1 and 2 was implemented. In each simulated scenario, nodes are randomly placed within a 700 by 50 m rectangle, a slight increase w.r.t. Bologna Central Station's dimensions. Then, all

[2] The RSSI is expressed in decibel-milliwatts (dBm)

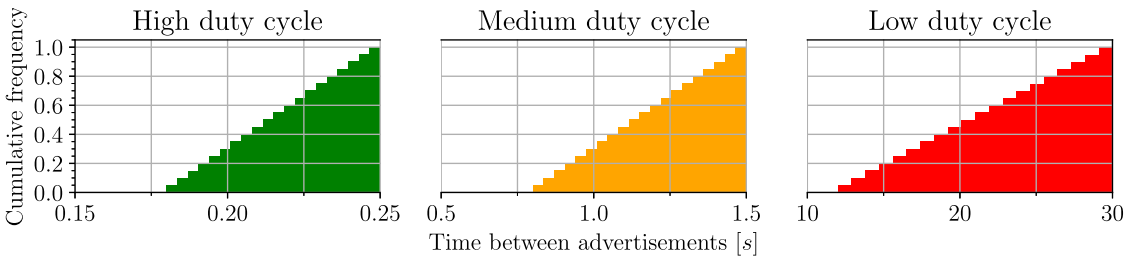


Fig. 5. Measured advertisement duty cycles.

Table 1
Chosen advertisement interarrival time (**t**) distributions.

BLE advertisement mode	Distribution
Low	$\mathbf{t} \sim \mathcal{U}(0.18, 0.25)$
Medium	$\mathbf{t} \sim \mathcal{U}(0.8, 1.5)$
High	$\mathbf{t} \sim \mathcal{U}(12, 30)$

Table 2
Measured occupancy compared with the number of BLE advertisements.

Occupancy	BLE Devices	Deviation	
		Value	%
214	216	+2	0.93%
227	225	−2	−0.88%
247	246	−1	−0.40%
231	226	−5	−2.20%
182	186	+4	2.20%
38	41	+3	7.90%

nodes start from an idle state and progress to the emergency mode after a random interval uniformly selected between 1 and 10 s. Afterwards, each node advertisement is propagated by sampling for each nearby ($\approx 120m$) peer the advertisement probability of reception calculated in Section 5.3.3. Each simulation ends after 600 s.

All simulations use the following algorithm parameters: $M = 10$, $N = 500$, $ACT = 1$, $T_{dt} = 3$ s, $T_{HM} = 10$ s, $T_{ML} = 30$ s. XXH3-64 is used, belonging to the xxHash family [47], as Bloom filter hash function h_i. Each hasher is initialized with $\frac{i}{K}\left(2^{64} - 1\right)$ as its seed ($i = 1, \ldots, K$), with K being the number of hashes calculated in Line 1 of Algorithm 1.

For each node count in $\{50, 100, 150, 200, 250, 300, 350, 400, 450, 500\}$, 500 instances were run with $BURST\_START$ set to 1 and 500 with it set to 0. Overall, 10,000 simulation were performed, for a total duration of approximately 1650 h.

5.3. Characterization of the real environment

5.3.1. BLE radio parameters

The operating system APIs do not provide an exact characterization of the duty cycle associated with each available advertisement mode (Low, Medium, High). Therefore, the BLE advertising interarrival time for each setting was measured, collecting data on approximately 1000 advertisements per mode. The assumption is that while this data is specific to the radio hardware of the tested device, similar patterns could be observed in other devices.

The gathered values, whose cumulative distribution function (CDF) is depicted in Fig. 5, show a distinct association between each mode and a uniformly distributed duty cycle. Higher modes are associated with shorter arrival rates. Specifically, the high rate is characterized by interarrival times between 0.18 and 0.25 s, the medium rate between 0.8 and 1.5 s, and the low rate by a comparatively longer period of 12 to 30 s.

Table 1 summarizes the overall identified duty cycle distributions, which will be used during the simulated portion of this study.

5.3.2. Ratio between BLE advertisements and occupancy

During this experiment, the goal was to determine the ratio between the number of BLE devices and the number of people in the case study area. To achieve this, BLE advertisements from individuals' phones participating in OFNs were passively monitored, provided those advertisements had "PHONE (512)" [48, 2.8.2] as their indicated class of device [40, 3.2.4]. To establish ground truth values, six distinct people counting sessions were conducted, each lasting as long as it took to traverse the entire station (≈ 10 min). During each session, the recording tablet and the security personnel were moved in unison to ensure synchronization between data measured by the application and the observed occupancy.

Table 2 gathers such measurements.

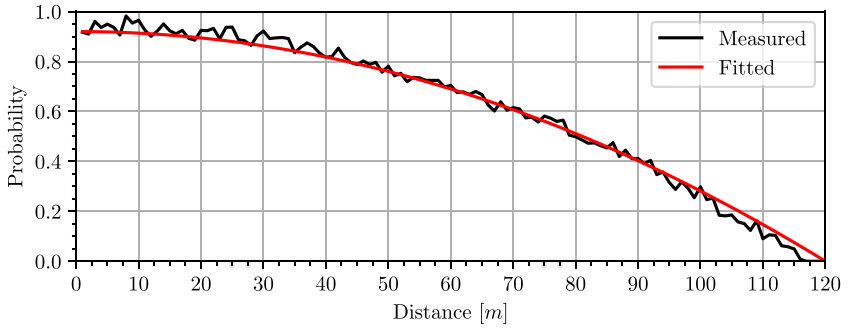


Fig. 6. Probability of reception for a BLE advertisement parameterized by distance.

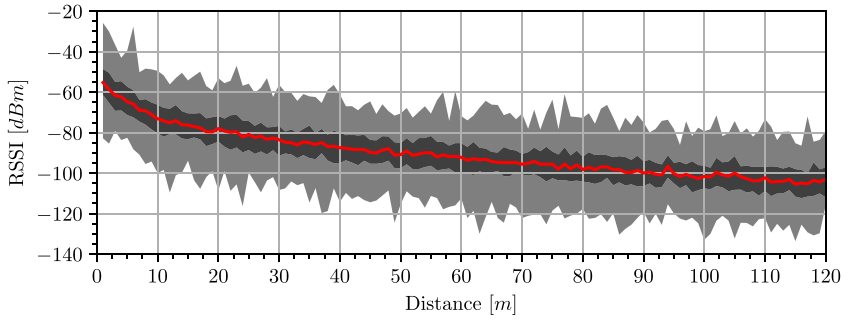


Fig. 7. Measured RSSI-distance relation.

Excluding the final scenario where the error was around 8%, nearly an equal number of BLE-capable phones to individuals was observed. This indicates that BLE has become a well-established technology. Moreover, these results suggest using the count of BLE devices as a reliable proxy for estimating the number of people present. This strong correlation suggests the viability of employing the proposed system to effectively count individuals in distress.

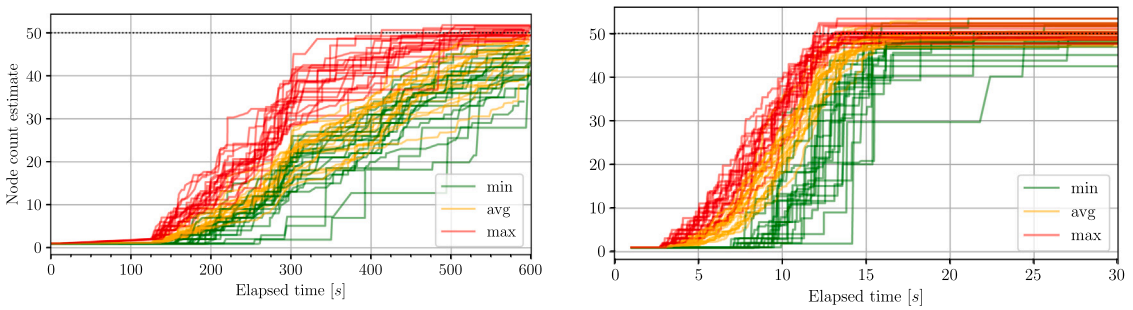
5.3.3. *Probability of reception for a given advertisement*

This experiment aims to determine the distance to the probability of reception relation for a BLE advertisement. Its raw measurements are processed by synchronizing the measurements of the two tablets together, calculating their mutual distance, binning them into 2.5 m intervals, and then assigning a probability by calculating how many advertisements were sent in the interval vs the ones actually received by the counterpart. In cases where the distance bin was unpopulated or contained less than 10 samples, linear interpolation was applied between the two adjacent bins. Fig. 6 depicts the observed trend, following approximately an exponential function. Starting from a distance of zero, where the probability of reception nears one, the probability decreases with distance, becoming almost negligible after the 105-m mark and reaching zero at around 120 m. Fig. 7 shows the RSSI envelope chart, indicating its median (red line), 50% percentile value range (darker shade), and entire range (lighter shade). By comparing these numbers with Fig. 6, lower advertisement receipt probabilities are predictably associated with lower signal strengths. A very broad RSSI envelope shape-wise was observed, possibly due to the many constructive and destructive interference phenomena impacting radio propagation. Worthy of notice, even though specific to the hardware used for the experiment, is the minimal recorded power of -135 dBm, associated with an extremely faint signal. This occasional reception of very low power advertisements is compatible with a remark from the Android documentation [35] saying that for a BLE radio set to "Aggressive mode", it is possible to "determine a match sooner, even with feeble signal strength and few number of sightings/match in a duration".

5.4. *Simulated trials*

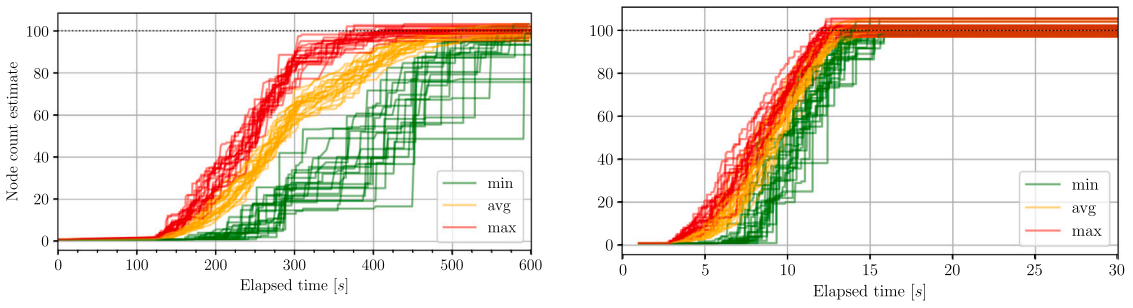
5.4.1. *Convergence time*

In addition to assessing the theoretical convergence in Theorem 1, the profile of convergence over time within the simulated scenarios is now examined. Figs. 8 to 17 illustrate the evolution of node count estimates as they change over time for varying network sizes with and without burst start optimization. The green line represents the minimum network size estimate across all nodes, while the orange line indicates the mean value, and the red line denotes the maximum network size estimate. As expected, the network node count estimates monotonically increase over time. Moreover, both the mean node count estimate and the distance between the minimum and maximum progressively decrease. This decreasing gap between the minimum and maximum estimates indicates convergence among the nodes of the network, as their individual estimates align more closely over time. When considering



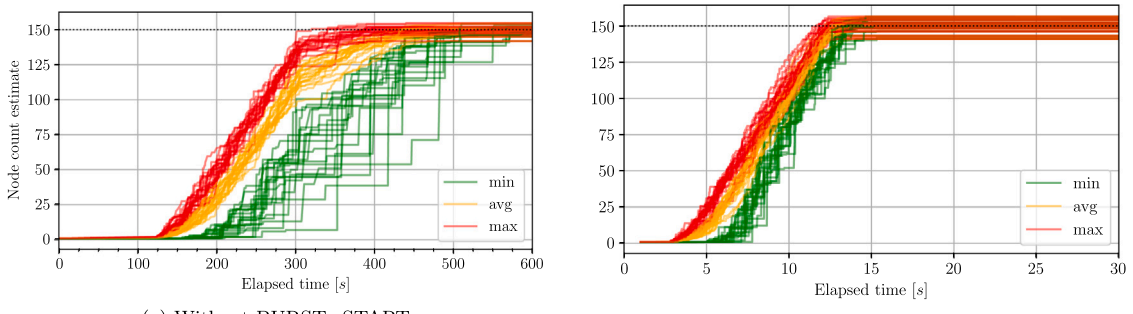
(a) Without BURST_START (b) With BURST_START

Fig. 8. Evolution of node count estimates for a network of size 50.



(a) Without BURST_START (b) With BURST_START

Fig. 9. Evolution of node count estimates for a network of size 100.



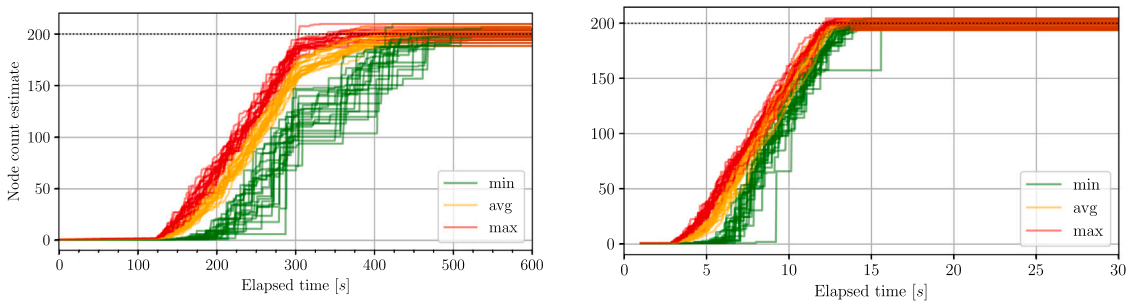
(a) Without BURST_START (b) With BURST_START

Fig. 10. Evolution of node count estimates for a network of size 150.

scalability in terms of the number of nodes, very sparse networks (such as the one with 50 nodes) experience longer convergence times due to each advertisement needing to traverse a longer path to reach every node. Conversely, as the simulation area remains constant, denser networks, i.e. the ones with more nodes, exhibit faster convergence times. This is because the average number of advertisements received per unit of time increases in denser networks, allowing for more rapid propagation of information throughout the system.

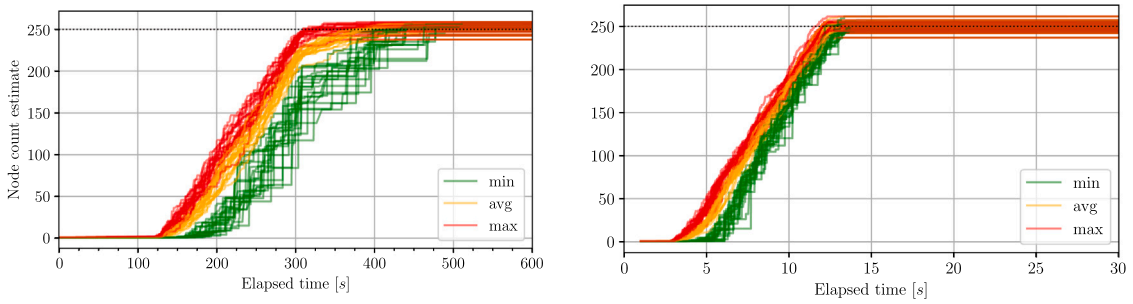
Table 3 provides an overview of the minimum, maximum, mean (μ), and standard deviation (σ) of convergence times. This time is defined as when all nodes in the simulation have reached the same value for their respective estimates.[3] The presented data shows that applying burst start optimization significantly reduces the time required for the network to achieve convergence, roughly by a factor of 10. Overall, the results show that with one exception (50 nodes and no BURST_START), all scenarios succeed in calculating a unified estimate within the allotted time of 10 min.

[3] Although the total number of nodes is constant, variations in the final estimation value can occur between different trials. For further insights on this matter, refer to the following section.



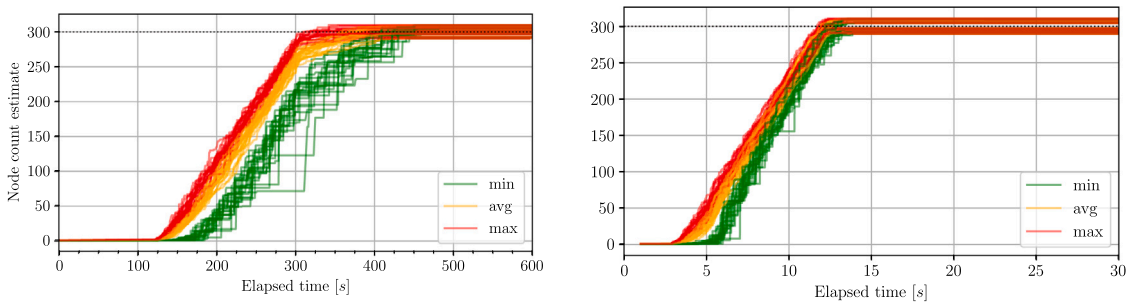
(a) Without BURST_START (b) With BURST_START

Fig. 11. Evolution of node count estimates for a network of size 200.



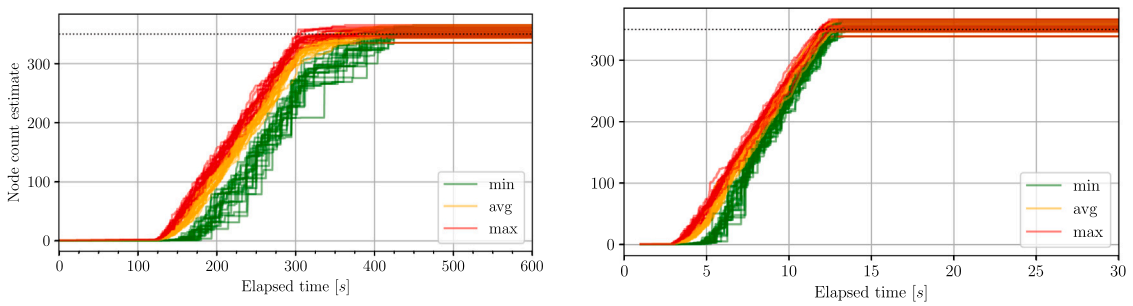
(a) Without BURST_START (b) With BURST_START

Fig. 12. Evolution of node count estimates for a network of size 250.



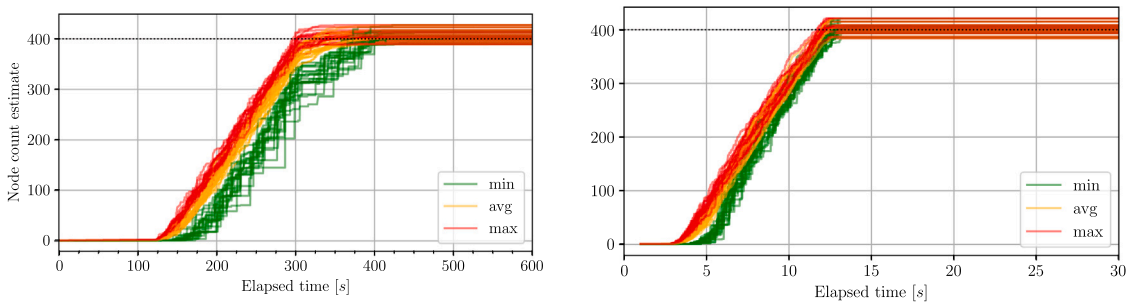
(a) Without BURST_START (b) With BURST_START

Fig. 13. Evolution of node count estimates for a network of size 300.



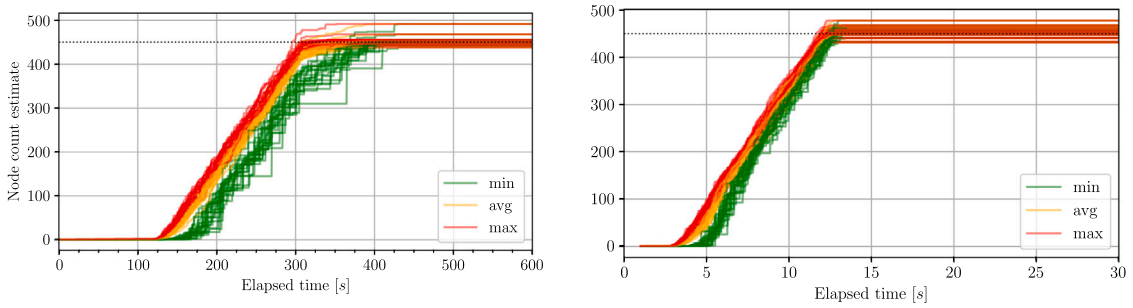
(a) Without BURST_START (b) With BURST_START

Fig. 14. Evolution of node count estimates for a network of size 350.



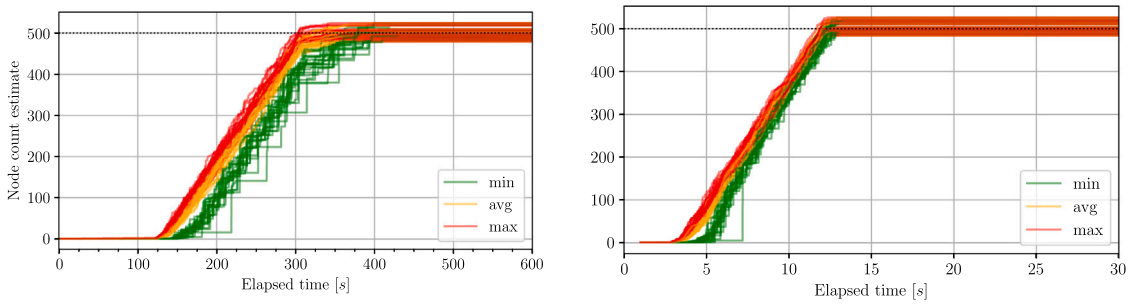
(a) Without BURST_START (b) With BURST_START

Fig. 15. Evolution of node count estimates for a network of size 400.



(a) Without BURST_START (b) With BURST_START

Fig. 16. Evolution of node count estimates for a network of size 450.



(a) Without BURST_START (b) With BURST_START

Fig. 17. Evolution of node count estimates for a network of size 500.

5.4.2. Accuracy

In the following, the accuracy of the proposed algorithm is analyzed by comparing the estimated count of devices with the actual number.

The error is defined as $e = \hat{N} - N$, where $\hat{N}$ represents the estimation calculated in Algorithm 1 at Line 23, and N is the actual value.

Table 4 depicts the minimum, mean (μ), maximum, and standard deviation (σ) of both absolute and percentage error values found across all trials involving a given number of nodes **N**.

The data shows that the algorithm keeps the percentage error under seven percent, with an overall average of around two percent. In some cases, the estimate comes extremely close to the real value, achieving error percentages of less than one percent. These figures suggest that the algorithm successfully provides a fairly accurate measurement of the actual size of the network.

Now, it can be checked whether the error exhibits a bias towards overestimating or underestimating the number of nodes by analyzing its values presented in Table 5.

It can be easily seen that the estimation error average hovers around zero. When looking at minimum and maximum errors, it can be observed that when making errors, the algorithm overestimation errors are higher relative to its underestimation errors in 7

Table 3
Effect of the BURST_START optimization to the convergence time for various network sizes (N).

N	BURST_START	Min [s]	Max [s]	μ [s]	σ [s]
50	0	failure	failure	failure	failure
	1	15.33	499.62	60.33	109.67
100	0	493.91	597.74	570.19	28.85
	1	13.87	17.98	15.34	1.03
150	0	483.77	599.46	538.76	39.36
	1	12.96	15.32	14.29	0.63
200	0	535.32	479.74	35.39	447.02
	1	13.51	15.58	14.04	0.52
250	0	418.67	510.77	461.72	27.41
	1	13.02	14.10	13.46	0.25
300	0	409.66	496.00	444.10	19.74
	1	13.05	13.95	13.40	0.25
350	0	387.46	450.81	422.18	16.72
	1	12.82	13.80	13.20	0.26
400	0	378.88	484.12	420.26	22.29
	1	12.57	13.47	13.05	0.22
450	0	388.97	433.33	412.15	14.55
	1	12.63	13.45	13.04	0.22
500	0	374.33	428.74	400.82	14.52
	1	12.56	13.19	12.86	0.17

Table 4
Network size estimation absolute ($|e|$) and percentage ($e_{\%}$) errors for various network sizes (N).

N	min $\|e\|$	$\mu(\|e\|)$	max $\|e\|$	$\sigma(\|e\|)$	min $e_{\%}$	$\mu(e_{\%})$	max $e_{\%}$
50	0.06	1.23	3.37	0.88	0.13	2.45	6.74
100	0.43	2.07	5.58	1.45	0.43	2.07	5.58
150	0.15	3.51	9.00	2.55	0.10	2.34	6.00
200	0.49	2.75	6.45	1.98	0.24	1.37	3.22
250	0.25	4.83	13.03	3.18	0.10	1.93	5.21
300	2.59	7.12	10.88	2.68	0.86	2.37	3.63
350	0.45	6.42	16.47	5.04	0.13	1.83	4.71
400	0.10	8.71	20.81	6.49	0.03	2.18	5.20
450	0.33	8.49	27.72	7.31	0.07	1.89	6.16
500	0.06	9.92	26.27	7.61	0.01	1.98	5.25

Table 5
Network size estimation error (e) for various network sizes (N).

N	min e	$\mu(e)$	max e	$\sigma(e)$
50	−2.41	−0.13	3.37	1.50
100	−2.90	0.69	5.58	2.43
150	−9.00	−0.29	6.53	4.32
200	−6.45	−0.95	4.30	3.25
250	−13.03	−0.10	11.60	5.78
300	−10.50	0.22	10.88	7.61
350	−11.11	3.09	16.47	7.55
400	−16.04	3.62	20.81	10.24
450	−18.98	1.64	27.72	11.09
500	−15.99	2.89	26.27	12.16

out of the 10 presented cases. This overestimation error matches the expectation: as the entire algorithm size estimation leverages Bloom filters as its underlying primitive, and since Bloom filters are prone to false positives, this is reflected in its estimation.

6. Performance, viability, security and privacy, threats to validity

6.1. Performance considerations and viability

The experimental data confirms the ability of the proposed solution to operate during the initial stages of an emergency, as its convergence time of less than one minute aligns with the goals outlined in Section 4.1. The formal evaluation presented in Theorem 1 confirms two key properties: first, the system allows access to the current estimate from any device on the network, and second, it will eventually converge. Continuing with the stated goals, the system is independent by construction from relying

on any externally-maintained communication infrastructure. Regarding efficiency, the system achieves high idle efficiency mainly due to operating system features enabling minimal overhead in its triggering mechanism. Active efficiency results from constraining the algorithm to operate within the available hardware capabilities, minimizing the time spent with the CPU in a high-power state. Notably, the power required for broadcasting the current status remains constant regardless of the number of nodes or network topology, offering a significant advantage over point-to-point approaches.

The practical viability of the proposed system is supported by satisfactory accuracy figures, suggesting its estimates could be a reasonably trustworthy aid to first responders. The experimental campaign measurements indicate that the majority of potential victims already possess devices capable of participating in the proposed system, thereby enabling this proposal to be rapidly deployed with minimally invasive changes. Moreover, the method does not present any immediate privacy risk – bar expensive electromagnetic techniques – which could be associated with its widespread adoption. Lastly, thanks to an end-to-end focus on efficiency, the system would have a minimal impact on battery life. These factors combined make the system a potentially viable candidate for real-world implementation to complement existing solutions providing assistance during emergency situations.

6.2. Security and privacy considerations

Privacy and security are critical aspects of any system designed for emergency scenarios, where protection of the individual and system reliability are essential. In this section, we provide an in-depth analysis of potential threats to our system and outline corresponding countermeasures.

6.2.1. Threat model

We consider adversaries with varying capabilities who may attempt to compromise the system's privacy and security:

- *Passive Adversaries.* Entities capable of eavesdropping on communications to collect transmitted data without altering it.
- *Active Adversaries.* Entities capable of injecting, modifying, or replaying messages to disrupt system operations or gain unauthorized access to information.

With this respect, note that the goal is to ensure that, even in the presence of such adversaries, the system protects users' privacy and maintains operational integrity.

To this end, three representative threats are identified, assuming adversaries have partial knowledge of the system's operations:

1. *Unauthorized Activation of the System.* Adversaries attempt to trigger the emergency counting system without a genuine emergency, aiming to access the node count estimate or cause unnecessary alarm.
2. *System Disruption through Data Tampering.* Adversaries inject false data or manipulate the counting algorithm to prevent convergence or artificially inflate the node count, rendering the system unreliable.
3. *Privacy Breach via Device Tracking.* Adversaries exploit the system's BLE broadcasts to track users' presence and movements, potentially infringing on personal privacy [49–51].

6.2.2. Countermeasures

In response to the outlined threats, a series of countermeasures can be implemented to enhance the security and privacy of the system. To prevent adversaries from triggering the system without an actual emergency, the system activates only in response to internally observable stimuli (i.e., not dependent on other devices' emergency statuses), it becomes challenging for an attacker to trick multiple devices into enabling the system without employing expensive hardware, such as a jammer capable of influencing the entirety of the targeted area.

To safeguard against data tampering and ensure the reliability of the node count, we observe that the counting algorithm uses a bitwise logical OR operation to aggregate hashed identifiers (HIDs). This method minimizes the influence of any single device on the overall count, reducing the impact of malicious data injection. Conversely, implementing full Byzantine fault tolerance does not seem to be a reasonable choice for resource-constrained BLE devices due to computational overhead [52,53]. Our aggregation method inherently limits the effectiveness of tampering by design.

Finally, to prevent adversaries from tracking devices and compromising user privacy, devices generate new RIDs for each activation of the system that are hashed using hash functions to produce HIDs. These HIDs are used in communications, guaranteeing that no PII is transmitted or can be inferred. As a matter of fact, the aggregation of HIDs via logical OR operations prevents the extraction of individual identifiers from the combined data, safeguarding against attempts to reverse-engineer user identities.

Limitations. Although we have discussed countermeasures for the identified security and privacy threats, some limitations must be acknowledged. The proposed threat model considers adversaries with limited knowledge and capabilities. More sophisticated attackers with advanced tools might find ways to disrupt the system or compromise privacy, such as by deploying widespread BLE signal jamming. Additionally, the system may be susceptible to denial of service attacks if adversaries flood the network with bogus BLE advertisements, potentially saturating the counting algorithm to its maximum value (see Eq. (3)).

6.3. Threats to validity

Construct validity. A concern that may affect the interpretation and validity of the results is that the use of Bloom filters introduces a probability of false positives, which may lead to overestimation of the node count. While our results indicate an average error of around 2%, specific scenarios could exhibit higher inaccuracies due to the probabilistic nature of the data structure. Signal propagation in non-line-of-sight environments presents another significant challenge, as demonstrated by our anecdotal experience in Bologna. The faint BLE signal proved incapable of traversing through thick reinforced concrete slabs, such as those separating the high-speed deepest part from the shopping area above. However, we observed that when devices were placed near (within 5 m) the escalator tunnels connecting different levels, advertisements could be relayed between them. Still, formulating a non location-specific penetration model is a significant challenge. Lastly, relying on the loss of network connectivity as the trigger for system activation may not reliably indicate an emergency. Users may experience connectivity loss for benign reasons, leading to false activations and potentially skewing data.

Internal validity. The initial phase of the experimental campaign was carried out using specific hardware: Lenovo TB-X605XL tablets equipped with Qualcomm Snapdragon 450 SoCs. The BLE performance characteristics of these devices may differ from those of other devices commonly used by the general population. Variations in BLE radio sensitivity, transmission power, and antenna design could affect the reception probabilities and, consequently, the convergence time and accuracy of the counting algorithm.

External validity. The experiments were conducted in a complex environment, Bologna Central Station, with specific architectural features and passenger flow patterns. This setting may not fully represent all potential emergency scenarios. Factors such as interference from other electronic devices, and the presence of large crowds could have influenced the BLE signal propagation and reception in ways not accounted for in the simulation.

7. Conclusions

In this paper, a novel approach to victim counting in post-disaster scenarios by leveraging a BLE-based spontaneous network was introduced. The method eliminates the need for external infrastructure, which can be compromised during disasters. The approach capitalizes on the widespread availability of mobile and wearable devices, reducing the need for additional specialized equipment and making the solution cost-effective and deployable. Extensive simulations and real-world measurements conducted in the challenging scenario of Bologna Central Station have demonstrated that the algorithm provides high accuracy with rapid convergence, even in large-scale scenarios. The system is designed to prioritize user privacy by using random identifiers (RIDs) and not storing or transmitting personal data. The method aligns with international emergency response frameworks by enabling accurate, quick, and efficient counting of affected individuals.

Looking ahead, future work will focus on further optimizing the algorithm, exploring direction-finding methods, and potentially integrating an exact counting mode that requires more power once aid arrives. Additionally, the application of this technology in very large-scale disasters involving thousands of devices will be studied.

CRediT authorship contribution statement

Giacomo Longo: Writing – review & editing, Writing – original draft, Software, Methodology, Investigation, Formal analysis, Conceptualization. **Alessandro Cantelli-Forti:** Writing – original draft, Validation, Methodology, Investigation, Conceptualization. **Enrico Russo:** Writing – review & editing, Writing – original draft, Validation, Supervision, Methodology, Investigation, Funding acquisition. **Francesco Lupia:** Writing – original draft, Validation, Investigation, Conceptualization. **Martin Strohmeier:** Writing – review & editing, Validation, Supervision. **Andrea Pugliese:** Writing – review & editing, Validation, Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was partially funded by the NextGenerationEU project “Security and Rights in CyberSpace” (SERICS). It was carried out while Giacomo Longo was enrolled in the Italian National Doctorate on Artificial Intelligence run by the Sapienza University of Rome in collaboration with the University of Genoa.

We extend our gratitude to the Ministry of the Interior’s Department of Public Security – Central Directorate for Road, Railway Police, and Special Units of the State Police, Railway Police Service – for granting the authorization for the experimental campaign, and to the Railway Police Department for Emilia Romagna for the assistance provided during the activities at the Bologna Central Railway Station.

Additionally, we appreciate the valuable suggestions from the Ministry of Interior of Italy - Department of Public Security, Office for Coordination and Planning of Police Forces - International Relations Service.

Data availability

Data will be made available on request.

References

[1] ITU-T. Focus Group on Disaster Relief Systems and Recovery, Technical Report on Telecommunications and Disaster Mitigation, Tech. Rep FG-DR&NRR, v 1.0, International Telecommunication Union, 2013.

[2] United Nations Office for Disaster Risk Reduction (UNDRR), Human cost of disasters: An overview of the last 20 years (2000–2019), 2020, URL https://www.undrr.org/publication/human-cost-disasters-overview-last-20-years-2000-2019. (Accessed: 6 July 2024).

[3] C. Bruch, R. Nijenhuis, S.N. McClain, 13 international frameworks governing environmental emergency preparedness and response: An assessment of approaches, in: The Role of International Environmental Law in Disaster Risk Reduction, Brill | Nijhoff, 2016, pp. 356–391, http://dx.doi.org/10.1163/9789004318816_014.

[4] United Nations Office for the Coordination of Humanitarian Affairs, Guidance Frameworks, URL https://asiadisasterguide.unocha.org/II-guidance-frameworks.html, United Nations Office for the Coordination of Humanitarian Affairs.

[5] United Nations Office for Disaster Risk Reduction, Sendai framework for disaster risk reduction 2015–2030, 2015, URL https://www.undrr.org/publication/sendai-framework-disaster-risk-reduction-2015-2030, United Nations Office for Disaster Risk Reduction.

[6] International Federation of Red Cross and Red Crescent Societies, Emergency Response Framework, International Federation of Red Cross and Red Crescent Societies, 2017.

[7] E. Gelenbe, G. Gorbil, Wireless networks in emergency management, in: Proceedings of the First ACM International Workshop on Practical Issues and Applications in Next Generation Wireless Networks, Mobicom '12, ACM, 2012, http://dx.doi.org/10.1145/2348714.2348716.

[8] F. Pervez, J. Qadir, M. Khalil, T. Yaqoob, U. Ashraf, S. Younis, Wireless technologies for emergency response: A comprehensive review and some guidelines, IEEE Access 6 (2018) 71814–71838, http://dx.doi.org/10.1109/access.2018.2878898.

[9] F. Mowbray, F. Mills, C. Symons, R. Amlôt, G. James Rubin, A systematic review of the use of mobile alerting to inform the public about emergencies and the factors that influence the public response, J. Contingencies Crisis Manag. 32 (1) (2023) http://dx.doi.org/10.1111/1468-5973.12499.

[10] J. Munoz-Castaner, P. Counago Soto, F. Gil-Castineira, F.J. Gonzalez-Castano, I. Ballesteros, A. di Giovanni, P. Colodron Villar, Your phone as a personal emergency beacon: A portable GSM base station to locate lost persons, IEEE Ind. Electron. Mag. 9 (4) (2015) 49–57, http://dx.doi.org/10.1109/mie.2015.2484922.

[11] A. Albanese, V. Sciancalepore, X. Costa-Perez, SARDO: An automated search-and-rescue drone-based solution for victims localization, IEEE Trans. Mob. Comput. 21 (9) (2022) 3312–3325, http://dx.doi.org/10.1109/tmc.2021.3051273.

[12] T. Řezník, B. Horáková, R. Szturc, Advanced methods of cell phone localization for crisis and emergency management applications, Int. J. Digit. Earth 8 (4) (2013) 259–272, http://dx.doi.org/10.1080/17538947.2013.860197.

[13] T. Li, J. Liang, Y. Ding, K. Zheng, X. Zhang, K. Xu, On design and performance of offline finding network, in: IEEE INFOCOM 2023 - IEEE Conference on Computer Communications, IEEE, 2023, http://dx.doi.org/10.1109/infocom53939.2023.10228880.

[14] A. Cantelli-Forti, M. Colajanni, S. Russo, Penetrating the silence: Data exfiltration in maritime and underwater scenarios, in: 2023 IEEE 48th Conference on Local Computer Networks, LCN, IEEE, 2023, pp. 1–6.

[15] L. Reichert, S. Brack, B. Scheuermann, A survey of automatic contact tracing approaches using bluetooth low energy, ACM Trans. Comput. Healthc. 2 (2) (2021) 1–33, http://dx.doi.org/10.1145/3444847.

[16] L. Maccari, V. Cagno, Do we need a contact tracing app? Comput. Commun. 166 (2021) 9–18, http://dx.doi.org/10.1016/j.comcom.2020.11.007.

[17] Y. Bengio, R. Janda, Y.W. Yu, D. Ippolito, M. Jarvie, D. Pilat, B. Struck, S. Krastev, A. Sharma, The need for privacy with public digital contact tracing during the COVID-19 pandemic, Lancet Dig. Health 2 (7) (2020) e342–e344, http://dx.doi.org/10.1016/s2589-7500(20)30133-3.

[18] H. Wen, Q. Zhao, Z. Lin, D. Xuan, N. Shroff, A study of the privacy of COVID-19 contact tracing apps, in: Security and Privacy in Communication Networks, Springer International Publishing, 2020, pp. 297–317, http://dx.doi.org/10.1007/978-3-030-63086-7_17.

[19] M. Matracia, M.A. Kishk, M.-S. Alouini, Comparing aerial-RIS- and aerial-base-station-aided post-disaster cellular networks, IEEE Open J. Veh. Technol. 4 (2023) 782–795, http://dx.doi.org/10.1109/OJVT.2023.3316117.

[20] M. Conti, M. Kumar, Opportunities in opportunistic computing, Computer 43 (1) (2010) 42–50, http://dx.doi.org/10.1109/MC.2010.19.

[21] A. Martín-Campillo, J. Crowcroft, E. Yoneki, R. Martí, Evaluating opportunistic networks in disaster scenarios, J. Netw. Comput. Appl. 36 (2) (2013) 870–880, http://dx.doi.org/10.1016/j.jnca.2012.11.001, URL https://www.sciencedirect.com/science/article/pii/S1084804512002275.

[22] Q. Ye, L. Cheng, M.C. Chuah, B.D. Davison, Performance comparison of different multicast routing strategies in disruption tolerant networks, Comput. Commun. 32 (16) (2009) 1731–1741, http://dx.doi.org/10.1016/j.comcom.2009.02.007, Special Issue of Computer Communications on Delay and Disruption Tolerant Networking, URL https://www.sciencedirect.com/science/article/pii/S0140366409000577.

[23] G. Chatzimilioudis, C. Costa, D. Zeinalipour-Yazti, W.-C. Lee, Crowdsourcing emergency data in non-operational cellular networks, Inf. Syst. 64 (2017) 292–302.

[24] G. Aloi, L. Bedogni, L. Bononi, O. Briante, M. Di Felice, V. Loscrì, P. Pace, F. Panzieri, G. Ruggeri, A. Trotta, STEM-NET: How to deploy a self-organizing network of mobile end-user devices for emergency communication, Comput. Commun. 60 (2015) 12–27.

[25] G. Aloi, O. Briante, M. Di Felice, G. Ruggeri, S. Savazzi, The SENSE-ME platform: Infrastructure-less smartphone connectivity and decentralized sensing for emergency management, Pervasive Mob. Comput. 42 (2017) 187–208, http://dx.doi.org/10.1016/j.pmcj.2017.10.004, URL https://www.sciencedirect.com/science/article/pii/S1574119217300214.

[26] R. Hasan, R. Hasan, T. Islam, Smart city technology for disaster management: Demonstrating the use of bluetooth low energy (BLE) beacons for emergency alert dissemination, in: 2022 IEEE 19th Annual Consumer Communications & Networking Conference, CCNC, 2022, pp. 931–932, http://dx.doi.org/10.1109/CCNC49033.2022.9700562.

[27] E. Kuada, B. Bannerman, Opportunistic rescue network for disaster management, in: 2017 IEEE AFRICON, 2017, pp. 917–922, http://dx.doi.org/10.1109/AFRCON.2017.8095604.

[28] T. Furutani, Y. Kawamoto, H. Nishiyama, N. Kato, Proposal and performance evaluation of information diffusion technique with novel virtual-cell-based wi-fi direct, IEEE Trans. Emerg. Top. Comput. 9 (3) (2021) 1519–1528, http://dx.doi.org/10.1109/TETC.2019.2891713.

[29] A. Hossain, S.K. Ray, R. Sinha, A smartphone-assisted post-disaster victim localization method, in: 2016 IEEE 18th International Conference on High Performance Computing and Communications; IEEE 14th International Conference on Smart City; IEEE 2nd International Conference on Data Science and Systems (HPCC/SmartCity/DSS), IEEE, 2016, pp. 1173–1179.

[30] Y. Wang, X. Yang, Y. Zhao, Y. Liu, L. Cuthbert, Bluetooth positioning using RSSI and triangulation methods, in: 2013 IEEE 10th Consumer Communications and Networking Conference, CCNC, 2013, pp. 837–842.

[31] M.E. Rida, F. Liu, Y. Jadi, A.A.A. Algawhari, A. Askourih, Indoor location position based on bluetooth signal strength, in: 2015 2nd International Conference on Information Science and Control Engineering, 2015, pp. 769–773, http://dx.doi.org/10.1109/ICISCE.2015.177.

[32] F. Brockmann, M. Handte, P.J. Marrón, CutiQueue: People counting in waiting lines using bluetooth low energy based passive presence detection, in: 2018 14th International Conference on Intelligent Environments, IE, 2018, pp. 1–8.
[33] A. Basalamah, Sensing the crowds using bluetooth low energy tags, IEEE Access 4 (2016) 4225–4233.
[34] T.D. Putri, T. Juhana, Mobile-openbts implementation of natural disaster victims search, in: 2017 3rd International Conference on Wireless and Telematics, ICWT, 2017, pp. 149–154, http://dx.doi.org/10.1109/ICWT.2017.8284157.
[35] T. Anugraha, K. Anwar, S.P. Jarot, Cellular communications-based detection to estimate location of victims post-disaster, in: 2019 Symposium on Future Telecommunication Technologies, Vol. 1, SOFTT, IEEE, 2019, pp. 1–5.
[36] T. Řezník, B. Horáková, R. Szturc, Advanced methods of cell phone localization for crisis and emergency management applications, Int. J. Digit. Earth 8 (4) (2015) 259–272.
[37] B.H. Bloom, Space/time trade-offs in hash coding with allowable errors, Commun. ACM 13 (7) (1970) 422–426, http://dx.doi.org/10.1145/362686.362692.
[38] S.J. Swamidass, P. Baldi, Mathematical correction for fingerprint similarity measures to improve chemical retrieval, J. Chem. Inf; Model; 47 (3) (2007) 952–964, http://dx.doi.org/10.1021/ci600526a, PMID: 17444629.
[39] S. Tarkoma, C.E. Rothenberg, E. Lagerspetz, Theory and practice of bloom filters for distributed systems, IEEE Commun. Surv. Tutor. 14 (1) (2012) 131–155, http://dx.doi.org/10.1109/SURV.2011.031611.00024.
[40] B.C.S.W. Group, Bluetooth Core Specification (amended), 2023, Version 5.4.
[41] H. Karvonen, C. Pomalaza-Ráez, K. Mikhaylov, M. Hämäläinen, J. Iinatti, Experimental performance evaluation of BLE 4 versus BLE 5 in indoors and outdoors scenarios, in: Advances in Body Area Networks I: Post-Conference Proceedings of BodyNets 2017, Springer, 2019, pp. 235–251.
[42] Apple and Google, Exposure notification - bluetooth specification, 2020.
[43] B. Brismar, L. Bergenwald, The terrorist bomb explosion in bologna, Italy, 1980: an analysis of the effects and injuries sustained, J. Trauma Acute Care Surg. 22 (3) (1982) 216–220.
[44] Fondazione Innovazione Urbana, Sezioni stazione di bologna centrale, 2015, https://www.fondazioneinnovazioneurbana.it/images/stories/nuovoallestimento/stazionesezioni.jpg. (Accessed 14 July 2024).
[45] Qualcomm, Snapdragon 450 mobile platform, 2024, URL https://www.qualcomm.com/products/mobile/snapdragon/smartphones/snapdragon-4-series-mobile-platforms/snapdragon-450-mobile-platform. (Accessed on 17 Jul 2024).
[46] Google, Google play services FusedLocationProviderClient, 2024, URL https://developers.google.com/android/reference/com/google/android/gms/location/FusedLocationProviderClient. (Last Accessed on 5 Jul 2024).
[47] Y. Collet, (Xxhash), 2019, URL https://github.com/Cyan4973/xxHash. (Accessed on 14 Jul 2024), version 0.8.2.
[48] Bluetooth S.I.G. Proprietary, Bluetooth assigned numbers, 2024, Version 2024-07-03.
[49] A. Thaljaoui, T. Val, N. Nasri, D. Brulin, BLE localization using RSSI measurements and iRingLA, in: 2015 IEEE International Conference on Industrial Technology, ICIT, IEEE, 2015, pp. 2178–2183.
[50] F. Buccafurri, V. De Angelis, C. Labrini, A privacy-preserving solution for proximity tracing avoiding identifier exchanging, in: 2020 International Conference on Cyberworlds, CW, 2020, pp. 235–242, http://dx.doi.org/10.1109/CW49994.2020.00045.
[51] H. Givehchian, N. Bhaskar, E.R. Herrera, H.R.L. Soto, C. Dameff, D. Bharadia, A. Schulman, Evaluating physical-layer ble location tracking attacks on mobile devices, in: 2022 IEEE Symposium on Security and Privacy, SP, IEEE, 2022, pp. 1690–1704.
[52] W. Zhong, C. Yang, W. Liang, J. Cai, L. Chen, J. Liao, N. Xiong, Byzantine fault-tolerant consensus algorithms: A survey, Electronics 12 (18) (2023) 3801.
[53] G. Zhang, F. Pan, Y. Mao, S. Tijanic, M. Dang'Ana, S. Motepalli, S. Zhang, H.-A. Jacobsen, Reaching consensus in the byzantine empire: A comprehensive review of bft consensus algorithms, ACM Comput. Surv. 56 (5) (2024) 1–41.