# RAMBLE: Opportunistic Crowdsourcing of User-Generated Data using Mobile Edge Clouds

Miguel Garcia[1], João Rodrigues[2], Joaquim Silva[3], Eduardo R. B. Marques[3], and Luís M. B. Lopes[3]
[1]*Critical Manufacturing, Portugal* (migafgarcia@gmail.com) [2]*Ericsson, Sweden* (joao.a.rodrigues@ericsson.com)
[3]*CRACS/INESC-TEC & Faculty of Sciences, University of Porto, Portugal* ({joaquim.silva,ebmarques,lmlopes}@fc.up.pt)

***Abstract*—We present RAMBLE[1], a framework for geo-referenced content-sharing in environments that have limited infrastructural communications, as is the case for rescue operations in the aftermath of natural disasters. RAMBLE makes use of mobile edge-clouds, networks formed by mobile devices in close proximity, and lightweight cloudlets that serve a small geographical area. Using an Android app, users ramble whilst generating geo-referenced content (e.g., text messages, sensor readings, photos, or videos), and disseminate that content opportunistically to nearby devices, cloudlets, or even cloud servers, as allowed by intermittent wireless connections. Each RAMBLE-enabled device can both produce information; consume information for which it expresses interest to neighboors, and; serve as an opportunistic cache for other devices. We describe the architecture of the framework and a case-study application scenario we designed to evaluate its behavior and performance. The results obtained reinforce our view that kits of RAMBLE-enabled mobile devices and modest cloudlets can constitute lightweight and flexible untethered intelligence gathering platforms for first responders in the aftermath of natural disasters, paving the way for the deployment of humanitarian assistance and technical staff at large.**

***Index Terms*—Crowd Sourcing, Edge Cloud, Mobile Computing**

## I. INTRODUCTION

Mobile phone usage has grown exponentially in recent years. Technology developments are also providing devices with more processing power, storage capacity, wireless communication technologies, and battery life. Today's devices are able to perform computationally intensive tasks, such as rendering graphics in games, playing high-quality video and performing image recognition [1]. They have also become a major source of data from the Internet periphery, feeding cloud infrastructures with vast amounts of data [2].

There are, however, scenarios in which applications cannot assume access to cloud-based servers. Natural disasters, such as hurricanes, earthquakes, volcanic eruptions tsunamis and large-scale fires, for example, can severely impair communication networks. In these scenarios, the importance of the communication capabilities of mobile devices and, eventually, local servers such as cloudlets [3], is paramount. Device-to-device (D2D) communication technologies, e.g., WiFi-Direct and Bluetooth, eventually combined with WiFi allow for sophisticated collaboration applications, commonly known as "crowdsourcing applications", to be implemented. In these applications, the data is generated, shared and eventually aggregated by the devices, without access to a cloud infrastructure. Such applications have been used with success in disaster scenarios where traditional communication infrastructures are commonly destroyed or severely impaired [4–7].

[1]ramble: (verb) wander about; travel aimlessly. (source: Thesaurus.com)

Inspired by these scenarios and the need to aggregate and depict geo-tagged and crowd-sourced information in them, e.g., as in [8, 9], we developed RAMBLE, a system for opportunistic generation, dissemination, and visualization of geo-tagged user-generated content. RAMBLE makes use of mobile edge clouds enabled by WiFi-Direct, lightweight proximity cloudlets that act as repositories and disseminators of user-collected data in small areas, plus, optionally, traditional cloud servers accessible via Internet. The shared data can be of any kind, but for disaster scenarios in particular, we focus on geotagged reports, text, sensor readings, audio and video messages, news, and map data would make sense to both to users in distress and rescue teams.

RAMBLE was the subject of the first author's Masters thesis [10], and was subsequently briefly described in a short abstract and poster [11]. This paper provides the first in-depth description and evaluation of this work, with the following main contributions. We describe the design and implementation of the RAMBLE data sharing framework, where devices discover other devices in a network neighborhood and synchronize data contents. We implemented mobile applications and servers (for cloudlets) that use the framework to synchronize a variety of contents: text messages, photos, and video. These instances use WiFi-Direct networking for short-term opportunistic device-to-device communication. Finally, we performed a real-world experiment as a proof-of-concept for the use of RAMBLE.

The experiment was conducted in Porto's Botanical Garden, and consisted of a crowd-sourced information gathering scenario with volunteer users carrying Android devices, along with the deployment of cloudlets. The results we obtained show that local networks of RAMBLE-enabled devices can effectively exchange their own or third party generated data either by device-to-device communication or also by cloudlet connections if available. The cloudlet tier, in our case formed by Linux-based Raspberry Pi computers with mesh and WiFi connectivity, significantly improves the dissemination of data by providing wider radio coverage and longer battery longevity. Collectively, these results support our view that kits of RAMBLE-enabled devices could easily be transported (in a small suitcase) to disaster areas in the immediate aftermath of the event and be used to gather local intelligence required

978-1-7281-7216-3/20/$31.00 ©2020 IEEE

for planning the deployment of humanitarian assistance and technical staff.

The remainder of the paper is structured as follows. Section II describes the design and implementation of RAMBLE. Section III describes the case-study deployment we used to assess the capabilities of RAMBLE app for disseminating geo-tagged, timestamped photos of Porto's Botanical Garden, and a corresponding analysis of the results. Section IV discusses related work. Finally, Section V puts forward the main conclusions and discusses future work.

## II. THE RAMBLE FRAMEWORK

### A. Overview

Figure 1 schematically illustrates a typical scenario for the use of RAMBLE, featuring three types of network peers: mobile devices, cloudlets and cloud servers. These peers can be connected through WiFi-Direct (for D2D communication), WiFi (device-to-cloudlet), mesh networking (cloudlet-to-cloudlet), and WiFi or 3G/4G Internet (cloud server communication). The goal is that user-generated content is disseminated during fortuitous connections among devices, or between devices and cloudlets or cloud servers. Connections will be volatile, e.g., as users roam in a within a region or when an Internet link allows communication with the cloud servers.

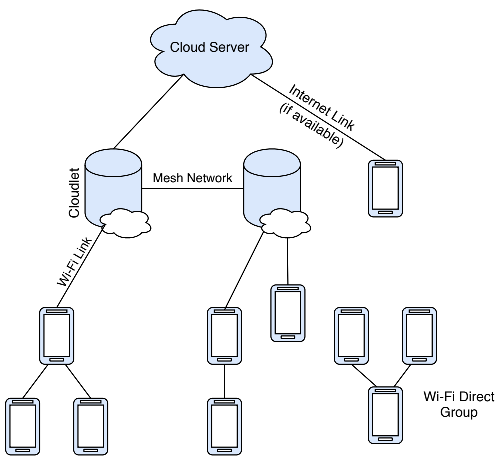
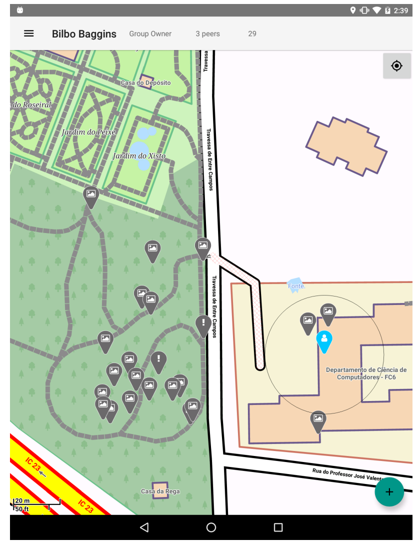


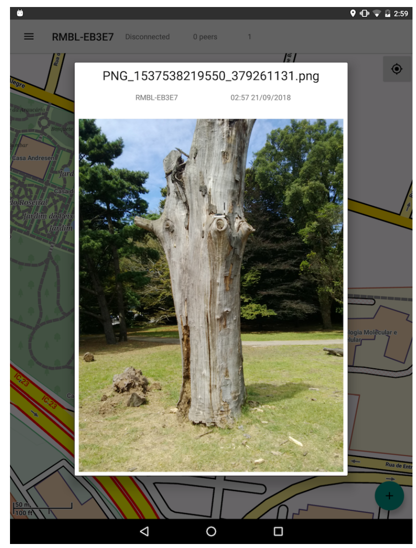

Fig. 1: A typical network scenario for RAMBLE.

RAMBLE is symmetric in the sense that each peer may act as a content generator, consumer, cache, and disseminator, with no pre-established hierarchy in terms of functionality. In the current version, we focus solely on mobile devices for the role of content generation, but data could also be injected directly and disseminated through cloudlets and cloud servers. This symmetry is made possible through a data dissemination service that is a core functionality shared by all peers and is discussed further on in this section.

For content generation and dissemination, users employ the RAMBLE Android app. Two screenshots of the app are shown in Figure 2. The main screen of the app (Figure 2a) displays a map view of all content stored in the peer, either generated locally or gathered through interactions with other peers. In the current version, offline map imagery from OpenStreetMap [12]




(a) Main screen.




(b) Content view.

Fig. 2: Screenshots of the RAMBLE Android app.

is pre-installed beforehand. The user is also informed (on top of the screen) of its current connectivity status and of the existence of nearby peers, and he is given the possibility of adding / generating new content (using the + button in the lower-right corner of the screen). The content markers may be clicked by the user to view the associated data (Figure 2b), which may take the form of photos, video, audio, or plain text.

### B. The RAMBLE service

The core service provided by RAMBLE is implemented at the level of each peer through a local SQLite database[2] and a Google RPC (gRPC) endpoint[3] for content dissemination over the network using remote procedure calls. The details of these

[2] https://sqlite.org
[3] https://grpc.io

components, namely, the database schema and a snippet of the gRPC specification, are illustrated in Figure 3.

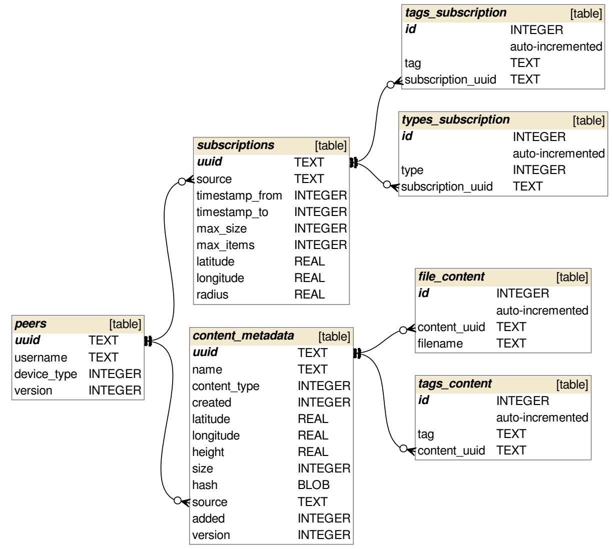
The database schema (Figure 3a) represents peers, subscription preferences, and content meta-data. To uniquely identify each type of information, we make use of universally unique identifiers (UUIDs) that are randomly generated at runtime. As peers are discovered in the network or their status changes, the corresponding information is stored to the database using the peers table: the name (username), type of device (device_type), and a content version counter (version) that is incremented by each peer when it generates content and when it gathers content from other peers. Each peer has an associated set of data subscription preferences, possibly changing over time, that specifies the kind of data it wishes to receive. Each subscription, an entry in the subscriptions table, may define several filters such as content origin (source), type (content_type), generation time (timestamp_from and timestamp_to), or geo-location ( latitude , longitude, and radius). The metadata of each content, an entry in the content_medatada table, has an associated name (name), type (content_type), source peer (source), timestamps for the original creation and insertion in the local database (created and added), geo-location ( latitude , longitude, and height), among other auxiliary attributes.

The gRPC specification (Figure 3b) defines the network interface for the service in terms of remote procedure calls between peers. The core interactions are defined by calls to hello and pull, shown at the bottom in the service section. These calls allow peers to probe for and synchronize content among themselves. Opportunistic encounters between peers lead to (symmetric) hello calls, that signal the intent of listing remote data according to each peer's subscription preferences, followed by corresponding calls to pull to obtain only new content (not stored already in the originating peer).

A call to hello from peer A to peer B allows A to obtain meta-data for content stored at B that matches the criteria specified in the call: the HelloRequest parameter encodes subscriptions of A, and the HelloResponse result encodes the corresponding matching content in B in terms of meta-data information (the Subscription and ContentMetaData messages are omitted in the snippet, as their structure mirrors the table structure of Figure 3a). A call to pull from A to B, performed only for an item not available yet in A, allows A to download the item stored at B: the PullRequest argument specifies the content UIID and the PullResponse result encodes the corresponding data that can be subsequently stored locally at A.

### *C. Support for mobile edge clouds*

In support of the RAMBLE service, specific types of edge clouds are formed. Android devices can form WiFi-Direct groups for D2D communication, seizing on chance encounters between users, and also access WiFi access points enabled by nearby cloudlets or infrastructural networks. Finally, cloudlets establish ad-hoc networks using the BATMAN protocol to communicate among themselves.




(a) Database schema.

```
message Header {
  Peer peer = 1;
  uint64 timestamp = 2;
}
message Chunk {
  ContentMetadata metadata = 1;
  bytes data = 2;
}
message HelloRequest {
  Header header = 1;
  repeated Subscription subscription = 2;
}
message HelloResponse {
  Header header = 1;
  repeated ContentMetadata content_metadata = 2;
}
message PullRequest {
  Header header = 1;
  string content_uuid = 2;
}
message PullResponse {
  Header header = 1;
  Chunk chunk = 2;
}
service Ramble {
  rpc hello(HelloRequest) returns (HelloResponse);
  rpc pull(stream PullRequest) returns (stream PullResponse);
}
```

(b) gPRC service specification (snippet).

Fig. 3: The RAMBLE core service.

WiFi-Direct groups or WiFi connections to access points are established by the RAMBLE Android app using the Hyrax middleware for mobile edge clouds [13, 14]. This middleware provides a convenient programming abstraction for D2D network formation using WiFi-Direct, but also connections to WiFi access points and the use of other communication technologies like Bluetooth. RAMBLE-enabled devices can form a WiFi-Direct group, with one device acting as group owner, meaning that the group owner essentially acts as a mobile access point, while also retaining the ability to be connected to a standard WiFi network. The process of Wifi-Direct network formation is a loop that runs on each device independently. The loop proceeds by first scanning the network for nearby WiFi-Direct groups or WiFi access points. If some

Fig. 4: Raspberry Pi devices used as cloudlets.

groups or access points are found, a connection is attempted with one of them. If no groups are found or connections successively fail, the device forms a WiFi-Direct group on its own, waiting for (up to 5) other devices to connect to it. The group is dismantled if it is empty for a maximum amount of time, and in that case the network formation loop restarts.

In the current implementation, RAMBLE cloudlets are Raspberry Pi 3 Mobel-B devices, shown in Figure 4, that can be easily deployed at an outdoor setting. The devices are powered using 20100 mAh TP-Link power bank for outdoor autonomy, and have two USB-attached D-Link DWA-172 WiFi cards with omnidirectional antennas. One of the WiFi cards is configured as an access point in the 2.4 GHz band using a distinct and non-overlapping 20 MHz channel per cloudlet. The other WiFi card is configured in mesh mode using the BATMAN protocol[4] over a 5 GHz band and a single channel (36) in all cloudlets with 20 MHz channel width. This scheme allows moving cloudlets in an outdoors setting or adding new ones on-the-fly, while the mesh network dynamically adapts to these changes.

## III. CASE-STUDY EXPERIMENT

### A. Setup

To evaluate RAMBLE, we conducted a real-world case-study experiment at Porto's Botanical Garden (Jardim Botânico do Porto). The experiment setup is sketched in Figure 5.

Ten volunteers were instructed to roam through the garden, carrying mobile devices running the RAMBLE Android app. We divided the area into four areas, identified A to D in the figure, such that users would begin their walk at a start location inside area A would move on to areas B, C and D, before returning to A to end their visit. To increase the chance of encounters between users, the latter were alternately instructed to walk through the garden clockwise or counter-clockwise in the designated areas, i.e., a path through A-B-C-D-A (white arrows) or through A-D-C-B-A (blue arrows). Users were also instructed to take photos at every point of interest (POI) encountered along their path. We selected several POIs in the days prior to the experiment. These included: trees identified with a species plate (only a small part of the trees in the garden are tagged in this way); two small lakes, and; two greenhouses. At the start of the experiment, we provided the

[4]https://www.open-mesh.org/projects/open-mesh/wiki

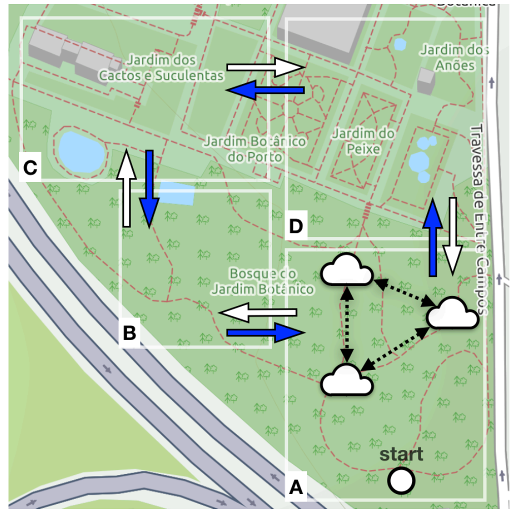



Fig. 5: Experiment setup.

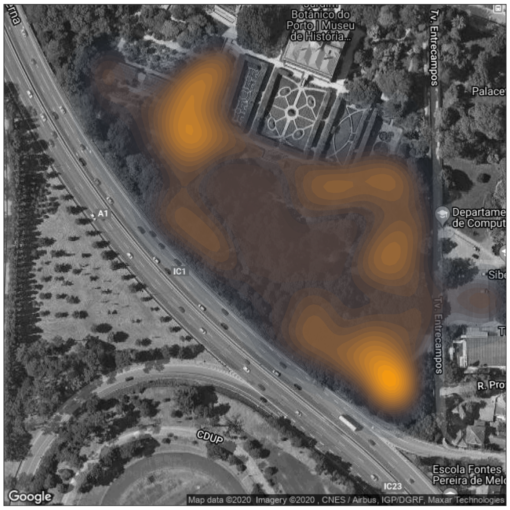
users with text descriptions and photos of the POIs so that they could recognize them. Users were instructed to take photos of a POI if no photo appeared in the Android app map view at the time of their visit. The aim was to approximate the behavior of a citizen science data collection campaign where users only generate content for yet uncovered POIs.

In the A area (Figure 5), we deployed a mesh of 3 Raspberry Pi cloudlets which, besides serving as data repositories, also provided WiFi access points to mobile devices. This meant that WiFi access by devices to cloudlets was especially good in this area, but also that, as users roamed away from area A the cloudlets' WiFi signal strength faded away, making data dissemination more likely to occur via WiFi-Direct. No other WiFi open access points existed in the area, hence connectivity through cloudlets or WiFi-Direct were the only means of exchanging data. The users carried Google Nexus tablets also without 3G/4G connectivity, starting with their battery fully charged. The RAMBLE app was pre-installed in the devices with a map of the area. A wildcard subscription was set for devices and cloudlets, i.e., meaning there were no filters set in the RAMBLE service for content dissemination. The aim was to get the maximum possible content dissemination out of the overall system.

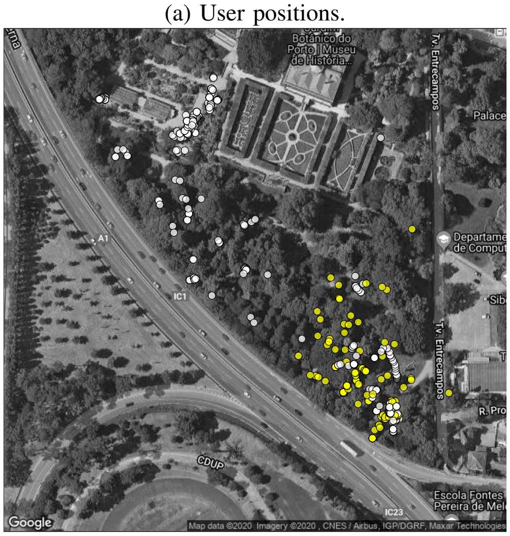
### B. Results

The elapsed time for the entire experiment was approximately 65 minutes. In total, 217 photos were created averaging 1.97 MB in size. These contents were exchanged 897 times between devices, including both user devices and cloudlets. We now analyze the main results from different perspectives.

*1) Spatial analysis:* Figure 6 provides an overview of the results derived from GPS and content transfer logs. A heat map is shown for user positions (6a), along with the locations for transfers initiated by a user device (6b) using either a WiFi connection to a cloudlet or as part of a WiFi-Direct group. The plots clearly indicate two areas where user movement and content transfers were more intensive: area A, in particular near the starting point (bottom left), and; area C,




(a) User positions.



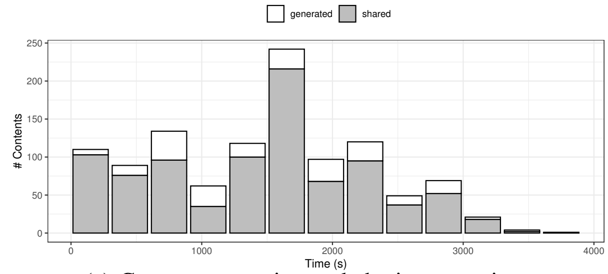

(b) Data transfers (WiFi-Direct: white; WiFi: yellow).

Fig. 6: Spatial overview of results.

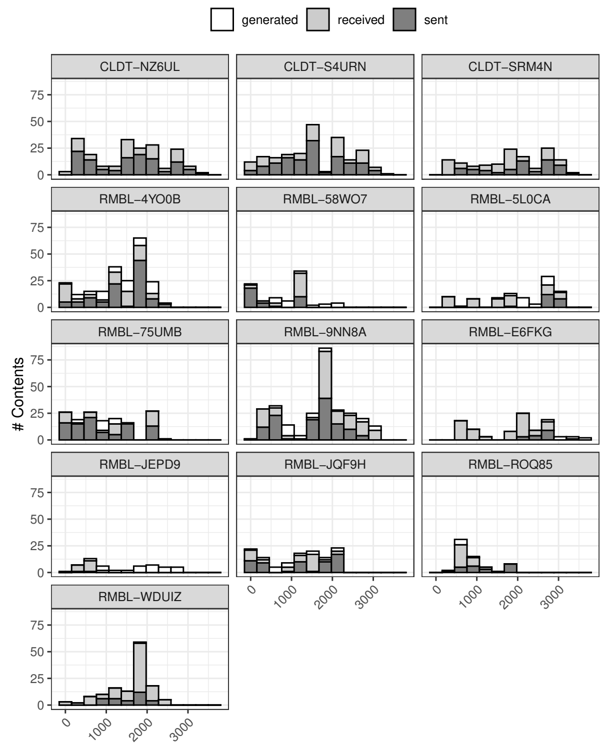
where users had more encounters due to the higher density of POIs. The other clear trait is that WiFi content transfers were basically limited to area A, where the cloudlets were deployed. Elsewhere, content transfer was enabled by WiFi-Direct.

*2) Temporal analysis:* Figure 7 depicts results for a temporal view of content generation and sharing (transfers from one peer to another) in 5 minute intervals: globally (7a); per device (7b), and; a summary for the activity of each peer during the entire experiment (7c). In the plots, as in others of this section, mobile devices and cloudlets are designated using identifiers of the form RMBL$-x$ and CLDT$-x$, respectively.

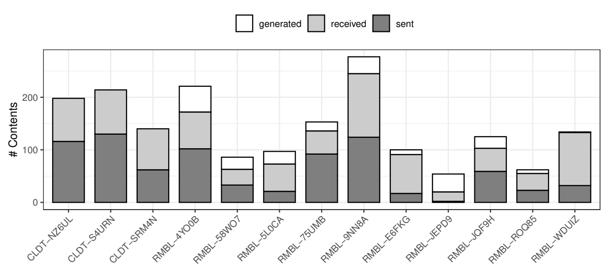
Globally, we can observe over time (7a) that contents are being shared among peers in significant volume relative to their generation. Hence, we should expect a reasonable rate of content dissemination among peers, an aspect that is examined in detail later in this section. A significant peak of activity is observed between $25\text{m}$ and $30\text{m}$ that is mostly related to an outburst of activity of device 9NN8A (7b). On the other hand, global activity waned in the last $15\text{m}$ of the experiment. In other periods, the activity tends to stay comparatively regular. Regarding individual peer behavior, we may observe (7b,7c) that cloudlets engaged in content transfers during the entire experiment and do not have strikingly different activity patterns, whereas devices exhibit heterogenous activity patterns in time, content generation, and content sharing. For




(a) Content generation and sharing over time.

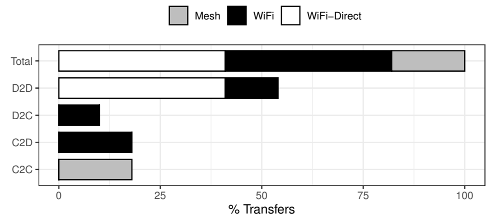



(b) Peer activity over time.

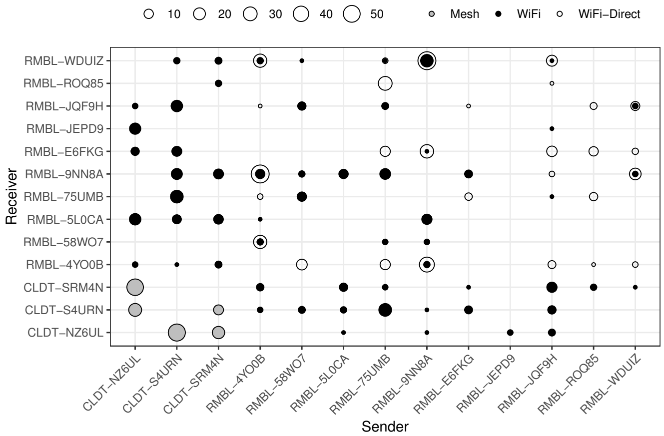



(c) Peer activity summary.

Fig. 7: Content generation and sharing over time.




(a) Transfers by type (%).




(b) Peer-to-peer transfers (# Contents).

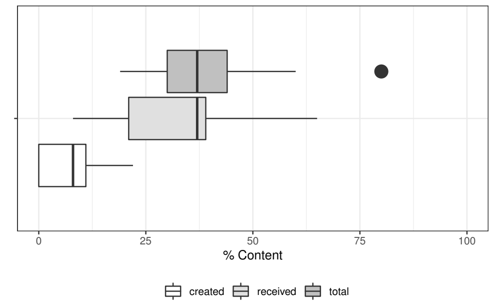
Fig. 8: Transfer analysis by connection type.

instance, we have high activity for devices 4YO0B and 9NN8A and, in the other extreme, comparatively very low activity for devices JEPD9 and ROQ85.

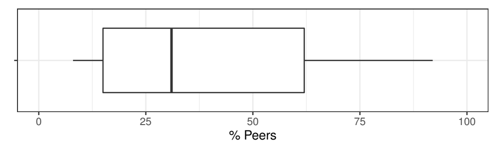
*3) Communications:* In Figure 8 we depict results regarding the use of communication links (Mesh, WiFi, and WiFi-Direct) for content transfers. We do it first in overall terms (8a), where we also distinguish the transfer flows: C2C: cloudlet-to-cloudlet; C2D: cloudlet-to-device; D2C: device-to-cloudlet, and; D2D: device-to-device. In addition, we provide a detailed view of communication links between peers (8b).

In overall terms, Figure 8a provides some clear insights. The share of transfers was $18\%$ for mesh networking, and an equal share of $41\%$ for both WiFi and WiFi-Direct. Hence, devices were involved in $82\%$ of all transfers (WiFi-Direct + WiFi), and the same number is $59\%$ for cloudlets (mesh + WiFi). From the perspective of content reception by devices, WiFi-Direct links were the most significant: the D2D WiFi-Direct transfer share ($41\%$ of global share) is 2.3 times higher than the WiFi C2D share ($18\%$), 3.2 higher than the D2D WiFi share ($13\%$), and 1.3 times higher than the combined WiFi C2D+D2D share ($31\%$). Considering the same perspective for cloudlets, mesh C2C and WiFi D2C transfer were equally significant: the global share is $18\%$ for both.

Looking at the peer-to-peer results in Figure 8b, the first visual insight is that there was a great deal of variety of "data encounters" between the universe of devices, without any immediate pattern of clustering or hierarchy. On average, each cloudlet interacted with 7.3 distinct devices, and each device




(a) Fraction of contents per device.




(b) Fraction of devices per content.

Fig. 9: Dissemination rates.

interacted with 5.3 other devices and 2.1 cloudlets. Other than that, for devices, the plot confirms the higher relative importance of WiFi-Direct for incoming content transfers (the upper part of the plot, that excludes cloudlet rows), and a great deal of heterogeneity (as previously in Figure 7). For cloudlets, a significant aspect is that we may observe that no transfers occurred from cloudlet S4URN to SRM4N; an inspection of the cloudlet logs revealed that this was due to a configuration bug in the BATMAN mesh.

*4) Content dissemination:* We consider two metrics to evaluate the success of content dissemination for the experiment, with results provided in Figure 9 in the form of box-plots: (9a) the fraction of all generated contents that were eventually stored (after being generated or received) by a device, and; (9b) symmetrically, per each content item, the fraction of devices that eventually store the item.

In terms of the content share per device, in Figure 9a we depict the total fraction, along with the individual contributions of locally generated and received content. Overall, the results are indicative of a significant level of content dissemination. The (aggregate) median value for all (device and cloudlet) peers is $37\%$ Moreover, we can also observe $75\%$ of peers stored approximately $30\%$ of all contents (1st quartile). We do not distinguish cloudlets and devices in the plots, but it is worth noting that the mean value is the same for cloudlets (and their individual share varies only by $2\%$), and decreases slightly to $32\%$ for devices. Finally, the depicted outlier in the figure refers to device 9NN8A (the most "active" one, as illustrated earlier), that managed to obtain $80\%$ of all contents (and comparatively, it only produced $15\%$ of all contents).

The results for dissemination share per content in Figure 9b are also indicative of reasonable levels of content dissemination. For instance, the median value is $31\%$ corresponding to $4$ (out of 13) peers, and $25\%$ of all contents reached at least $8$ peers (the 3rd quartile is $62\%$).

*5) Battery consumption:* Finally, we provide results for the battery consumption of the RAMBLE Android app during the experiment, where time is relative to the absolute start time per each device. The individual battery levels per device over time were sampled using the Android API for that purpose[5], and are shown in Figure 10. We can observe that the battery consumption was quite significant. For instance, the battery was depleted by $50\%$ after one hour for device 5L0CA, and by close to $30\%$ for devices 75UMB and JEPD9 over roughly the same amount of time.

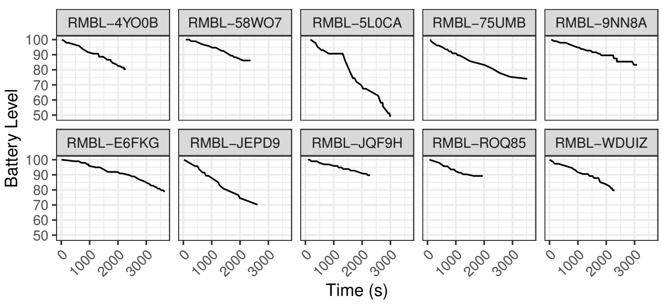



Fig. 10: Battery consumption of Android devices.

In the current prototype, we made no special provisions to optimize the app in terms of energy consumption, and our analysis of the results is limited in this paper. We believe that the continuous use of WiFi/WiFi-Direct plus GPS explains the battery drainage. In particular, the WiFi-Direct group scanning/formation loop should have a significant impact, even more than the process of sharing data itself: very active peers in data sharing like 9NN89 (cf. Figure 7) consumed little energy compared with peers with low data sharing but high energy consumption like JEPD9 (the latter may for instance have spent more effort/energy in network formation). We need to examine all these factors in detail, and also consider alternatives that should lower battery consumption in principle, e.g., use Bluetooth LE for device discovery before attempting WiFi-Direct group formation, or use plain Bluetooth in place of WiFi-Direct at the cost of lower bandwidth or only for transferring contents with limited size (e.g., text messages, low-resolution photos).

A similar analysis would also be required for the Raspberry Pi cloudlets in our setup, but we could not configure the necessary battery monitoring in time for the experiment. Results from a previous experiment that used the same cloudlet setup (Raspberry Pi models, battery supply, network cards and BATMAN meshing), but with significantly higher traffic demands, indicate that the cloudlets can operate for periods of at least 3 hours without recharging [15].

## IV. RELATED WORK

RAMBLE comes in sequence to our previous work in mobile edge clouds, such as the Hyrax middleware for network formation and communication [13, 14] that we used in RAMBLE for WiFi-Direct communications, a case-study experiment concerning video dissemination during a sport events where we made use of the same network architecture as that of RAMBLE [15], and a framework for adaptive computation offloading [16] where we similarly consider several network tiers (devices, cloudlets, and cloud servers).

Mobile crowd-sourcing apps are nowadays routinely used, particularly in real-world disaster scenarios [5, 6]. Examples of widely used tools include: Ushahidi [17], a open-source platform that gathers crowd-sourced information by several means (e.g., SMS, email, Twitter), and has been used for more than 10 years in several contexts, e.g., earthquake [4] events; FrontlineSMS [18, 19], that solely makes use of SMS messages for crowd-sourced information, and can be used in combination with Ushahidi [20], and; OpenStreetMap [12], the crowd-sourcing mapping platform, is also an important tool in disaster scenarios for collaborative mapping [4, 8, 9].

Mobile edge clouds can be important assets in this context. Cloudlet-based infrastructures are advocated as a means of rapidly provisioning services in disaster scenarios [3, 21]. Similarly to the use of Rasbperry Pi in RAMBLE, other works considered embedded devices that function as cloudlets e.g., wireless routers [22] or smartphones [23] (note that a RAMBLE-enabled Android device can also be considered as a mobile cloudlet). Also as in RAMBLE, ad-hoc mesh networks are used for instance in Twimight [24, 25], where Twitter messages can be exchanged in "disaster mode", the Serval project [26] that lets users talk and exchange geo-referenced media content and has been employed in several real-world tests [27], and real-world community networks like Guifi.net [28] whose nodes can provide cloud services through the Cloudly Linux distribution [29]. Finally, the use of WiFi-Direct or Bluetooth, which can be used off-the-shelf in modern smartphones (no device "rooting" or special hardware is usually required), enables systems like ProximAid [30], Emergency Direct Mobile App [31], and also commercial apps like FireChat [32].

## V. CONCLUSION

In this paper we presented RAMBLE, a system for opportunistic dissemination of content designed with infrastructure deprived environments in mind. Each instance of RAMBLE, running on a mobile device or a cloudlet, features a database that acts as a local source of truth and a discovery service that can be used to find other RAMBLE-enabled peers in the neighborhood. Once found, peers use a custom protocol to establish short-term opportunistic contacts aimed at synchronizing their local databases. Data typically includes timestamped, geo-tagged, text messages, sensor readings, photos or videos. We implemented the framework and provided its functionalities in the form of an API and an Android app that allows users to generate data and gather data from other users when the devices are in connectivity range.

RAMBLE was evaluated during a real-world experiment at Porto's Botanical Garden in which users collected and exchanged data using the Android app. The analysis of the logs

[5]https://developer.android.com/reference/android/os/BatteryManager

shows that the framework is capable of very effectively disseminating contents between participating peers, while seamlessly switching between three different wireless protocols. These results suggest that our initial motivation for this work was right: modest setups of mobile devices and cloudlets with appropriate software can be used as simple, portable and effective communication infrastructures to collect local intelligence in the aftermath of natural disasters.

Future work will focus on the energy efficiency of the framework since the results do show significant battery consumption during the experiments, mostly a result of wireless communication. Also of interest is the possibility that peers may perform some in loco pre-processing of the data, besides disseminating it. Photos and videos, for example, could be analyzed using machine learning techniques [33, 34] to identify specific objects or characteristics, thus allowing a synopsis of their contents to be disseminated instead. This pre-processing could be performed by the peer that generates the data or, as we consider in [16], be offloaded to a local cloudlet with more computational power and henceforth the corresponding synopsis disseminated to other peers.

## References


[1] N. Fernando, S. W. Loke, and W. Rahayu, "Mobile Cloud Computing: A Survey," *Future Generation Computing Systems.*, vol. 29, no. 1, pp. 84–106, 2013.

[2] U. Drolia, R. Martins *et al.*, "The Case for Mobile Edge-Clouds," in *Proc. Usenix Annual Technical Conference (UIT/ATC)*. IEEE, 2013.

[3] M. Satyanarayanan, G. Lewis *et al.*, "The role of cloudlets in hostile environments," *IEEE Pervasive Computing*, vol. 12, no. 4, pp. 40–49, 2013.

[4] M. Zook, M. Graham *et al.*, "Volunteered geographic information and crowdsourcing disaster relief: a case study of the haitian earthquake," *World Medical & Health Policy*, vol. 2, no. 2, pp. 7–33, 2010.

[5] M. Poblet, E. García-Cuesta, and P. Casanovas, "Crowdsourcing tools for disaster management: A review of platforms and methods," in *AI Approaches to the Complexity of Legal Systems*. Springer, 2014, pp. 261–274.

[6] M. L. Tan, R. Prasanna *et al.*, "Mobile applications in crisis informatics literature: A systematic review," *International Journal of Disaster Risk Reduction*, vol. 24, pp. 297–311, 2017.

[7] H. Yamamura, K. Kaneda, and Y. Mizobata, "Communication problems after the great East Japan earthquake of 2011," *Disaster medicine and public health preparedness*, vol. 8, no. 4, pp. 293–296, 2014.

[8] T. H. Poiani, R. dos Santos Rocha *et al.*, "Potential of collaborative mapping for disaster relief: A case study of OpenStreetMap in the Nepal earthquake 2015," in *Proc. Hawaii International Conference on System Sciences (HICSS)*. IEEE, 2016, pp. 188–197.

[9] R. Soden and L. Palen, "From crowdsourced mapping to community mapping: The post-earthquake work of OpenStreetMap Haiti," in *Proc. International Conference on the Design of Cooperative Systems (COOP)*. Springer, 2014, pp. 311–326.

[10] M. Garcia, "Ramble: opportunistic content dissemination for infrastructure-deprived environments," Master's thesis, University of Porto, 2018.

[11] M. Garcia, J. Rodrigues *et al.*, "Ramble: opportunistic content dissemination for infrastructure-deprived environments," in *INForum'18*, 2018, poster/short abstract.

[12] M. Haklay and P. Weber, "Openstreetmap: User-generated street maps," *IEEE Pervasive Computing*, vol. 7, no. 4, pp. 12–18, 2008.

[13] J. Rodrigues, E. R. B. Marques *et al.*, "Towards a middleware for mobile edge-cloud applications," in *Proc. Workshop on Middleware for Edge Clouds & Cloudlets (MECC)*. ACM, 2017, pp. 1–6.

[14] J. Rodrigues, "A middleware for mobile edge-cloud applications," Ph.D. dissertation, University of Porto, 2019.

[15] J. Rodrigues, E. R. B. Marques *et al.*, "Video dissemination in untethered edge-clouds: a case study," in *Proc. International Conference on Distributed Applications and Interoperable Systems (DAIS)*. Springer, 2018, pp. 137–152.

[16] J. Silva, E. R. B. Marques *et al.*, "Jay: Adaptive Computation Offloading for Hybrid Cloud Environments," in *Paper draft submitted to FMEC'2020*, 2020.

[17] O. Okolloh, "Ushahidi, or 'testimony': Web 2.0 tools for crowdsourcing crisis information," *Participatory learning and action*, vol. 59, no. 1, pp. 65–70, 2009.

[18] K. Banks, S. M. McDonald, and F. Scialom, "Mobile technology and the last mile: "Reluctant innovation" and FrontlineSMS," *Innovations: Technology, Governance, Globalization*, vol. 6, no. 1, pp. 7–12, 2011.

[19] S. L. Bel, D. Chavernac *et al.*, "FrontlineSMS as an Early Warning Network for Human-Wildlife Mitigation: Lessons Learned from Tests Conducted in Mozambique and Zimbabwe," *The Electronic Journal of Information Systems in Developing Countries*, vol. 60, no. 1, pp. 1–13, 2014.

[20] K. Banks and E. Hersman, "FrontlineSMS and Ushahidi - a demo," in *Proc. International Conference on Information and Communication Technologies and Development (ICTD)*. IEEE, 2009, pp. 484–484.

[21] G. A. Lewis, S. Echeverría *et al.*, "Cloudlet-based cyber-foraging for mobile systems in resource-constrained edge environments," in *Companion Proc. International Conference on Software Engineering (ICSE)*. ACM, 2014, p. 412–415.

[22] C. Meurisch, T. A. B. Nguyen *et al.*, "Upgrading wireless home routers as emergency cloudlet and secure DTN communication bridge," in *Proc. International Conference on Computer Communication and Networks (ICCCN)*. IEEE, 2017, pp. 1–2.

[23] E. Koukoumidis, D. Lymberopoulos *et al.*, "Pocket cloudlets," *ACM SIGPLAN Notices*, vol. 46, no. 3, pp. 171–184, 2011.

[24] F. Legendre, T. Hossmann *et al.*, "30 years of wireless Ad Hoc networking research: what about humanitarian and disaster relief solutions? What are we still missing?" in *Proc. International Conference on Wireless Technologies for Humanitarian Relief (ACWR)*, 2011, pp. 217–217.

[25] T. Hossmann, P. Carta *et al.*, "Twitter in Disaster Mode: Security Architecture," in *Proc. Special Workshop on Internet and Disasters*. ACM, 2011.

[26] P. Gardner-Stephen and S. Palaniswamy, "Serval mesh software-WiFi multi model management," in *Proc. International Conference on Wireless Technologies for Humanitarian Relief (ACWR)*, 2011, pp. 71–77.

[27] P. Gardner-Stephen, R. Challans *et al.*, "The Serval Mesh: A platform for resilient communications in disaster & crisis," in *Proc. Global Humanitarian Technology Conference (GHTC)*. IEEE, 2013, pp. 162–166.

[28] R. Baig, R. Roca *et al.*, "Guifi.net, a crowdsourced network infrastructure held in common," *Computer Networks*, vol. 90, pp. 150–165, 2015.

[29] M. Selimi, A. M. Khan *et al.*, "Cloud services in the Guifi.net community network," *Computer Networks*, vol. 93, pp. 373–388, 2015.

[30] K. Koumidis, P. Kolios *et al.*, "ProximAid: Proximal adhoc networking to aid emergency response," in *Proc. International Conference on Information and Communication Technologies for Disaster Management (ICT-DM)*. IEEE, 2015, pp. 20–26.

[31] M. Di Felice, L. Bedogni, and L. Bononi, "The Emergency Direct Mobile App: Safety Message Dissemination over a Multi-Group Network of Smartphones Using Wi-Fi Direct," in *Proc. International Symposium on Mobility Management and Wireless Access (MobiWac)*. ACM, 2016, p. 99–106.

[32] N. Cohen, "Hong Kong protests propel FireChat phone-to-phone app," *The New York Times*, vol. 5, pp. 26–28, 2014.

[33] K. Ota, M. S. Dao *et al.*, "Deep learning for mobile multimedia: A survey," *ACM Trans. on Multimedia Computing, Communications, and Applications (TOMM)*, vol. 13, no. 3s, p. 34, 2017.

[34] M. Xu, J. Liu *et al.*, "A first look at deep learning apps on smartphones," in *Proc. WWW*. ACM, 2019, pp. 2125–2136.


**Acknowledgements.** This work was partially funded by project SafeCities (POCI-01-0247-FEDER-041435) from Fundo Europeu de Desenvolvimento Regional (FEDER), through COMPETE 2020 and Portugal 2020.