Improving Emergency Preparedness and Response in Rural Areas

Karyn Doke
Department of Computer Science
University at Albany, SUNY
kdoke@albany.edu

Habib O. Affinnih
Department of Computer Science
University at Albany, SUNY
haffinnih@albany.edu

Qianli Yuan
Rockefeller College of Public Affairs and Policy
University at Albany, SUNY
qyuan@albany.edu

Mila Gasco
Center for Technology in Government and Rockefeller College of Public Affairs and Policy
University at Albany, SUNY
mgasco@albany.edu

Jose Ramon Gil-Garcia
Center for Technology in Government and Rockefeller College of Public Affairs and Policy
University at Albany, SUNY
jgil-garcia@albany.edu

Petko Bogdanov
Department of Computer Science
University at Albany, SUNY
pbogdanov@albany.edu

Mariya Zheleva
Department of Computer Science
University at Albany, SUNY
mzheleva@albany.edu

ABSTRACT

The unique socio-economic structure of rural communities makes them particularly vulnerable to emergencies. However, rural emergency preparedness and response (EPR) significantly lag behind their urban counterparts. A key obstacle to timely dissemination of emergency information is limited broadband, which in turn limits agencies' abilities to (i) disseminate preparedness and response information to residents and (ii) coordinate in the face of a disaster.

We aim to improve rural EPR services by aggregating information from national, state and county-based sources and disseminating it in rural communities with limited broadband by leveraging first responders' and residents' mobility. To this end, we co-design and develop an emergency smartphone app (**EApp**) in collaboration with a rural community in New York State. We study **EApp**'s performance in the lab and through deployments, focusing on energy use, required socio-physical interactions and timeliness of information access. Our findings elucidate critical limitations of off-the-shelf Android platforms to support hands-free opportunistic networks. To address these limitations, we design protocols on top of Wi-Fi direct enabling near 100% success rate in peer-to-peer (P2P) information exchange. Our results inform an optimal end-to-end design and deployment of a rural P2P information dissemination platform.

COMPASS '21, June 28-July 2, 2021, Virtual Event, Australia

ACM ISBN 978-1-4503-8453-7/21/06…$15.00
https://doi.org/10.1145/3460112.3471944

CCS CONCEPTS

• **Networks → Peer-to-peer networks**; **Network performance evaluation**; **Network experimentation**; **Mobile networks**.

ACM Reference Format:
Karyn Doke, Habib O. Affinnih, Qianli Yuan, Mila Gasco, Jose Ramon Gil-Garcia, Petko Bogdanov, and Mariya Zheleva. 2021. Improving Emergency Preparedness and Response in Rural Areas. In *ACM SIGCAS Conference on Computing and Sustainable Societies (COMPASS) (COMPASS '21), June 28-July 2, 2021, Virtual Event, Australia.* ACM, New York, NY, USA, 13 pages. https://doi.org/10.1145/3460112.3471944

1 INTRODUCTION

Natural and man-made disasters are claiming the lives and the livelihoods of people across the globe. Rural areas are particularly susceptible to such devastating events due to their wide geographical spread, limited access to information technology and shortage of emergency preparedness and response (EPR) personnel [3, 52]. Modern EPR services rely increasingly on technological tools and mobile Internet [5], however, rural broadband is still largely unavailable with some 15.2 million (30%) of rural U.S. residents still lacking access [1]. *These factors hamper rural communities' resilience to emergencies and disasters, and collectively underpin the emergency preparedness and response digital divide.*

Effective EPR requires access to both static instructional information (e.g. first aid/resuscitation tutorials [4]) and dynamic alerts pertaining to unfolding emergencies. To receive information from these outlets, users are expected to proactively subscribe to the information sources and have continuous Internet access. While these requirements can be met in the urban context where mobile broadband is ubiquitous and the population is technically well-versed, the same cannot be readily assumed for rural residents.

Message passing applications for disaster response based on opportunistic mesh networks have been developed for areas without

Internet access [46, 50]. While such advances tackle multi-hop routing from a source (e.g. victim) to a destination (e.g. responder), they often assume that mesh connections are continuously available via peer-to-peer (P2P) communication [30, 34, 42, 44, 47]. However, as we demonstrate in this paper, the practical realization of opportunistic smartphone-enabled networks dependent on human mobility is not trivial and poses research challenges for information dissemination.

In this paper, we investigate the key technological challenges for bridging the gap in EPR information dissemination in rural areas with intermittent or no Internet connectivity. To this end, we partner with a rural community and agencies in Upstate New York to co-design a mobile smartphone application called the **EApp**. The **EApp** aggregates emergency information from multiple federal, state and county official outlets including FEMA IPAWS [16], NOAA [23], NY Alert [20], 511NY [21], and emergency-related social media pages. Together, these sources provide alerts on fires, transportation, weather, hazardous materials, medical emergencies and others identified by our community partners. **EApp** subscribers receive the aggregated EPR information through an intuitive mobile user interface. When clients have Internet access, they pull information directly from the **EApp** server. When offline, residents can pull new alerts from community members or first responders through opportunistic P2P exchange. The design of **EApp**'s P2P functionality is geared towards wide applicability: we focus on stock Android Wi-Fi Direct [28] running on low-cost mobile devices[1]. We investigate and design solutions for the challenges arising due to this limited software-hardware environment and demonstrate that it provides sufficient range and throughput for rural EPR information dissemination.

Our work makes the following contributions:

- We design, develop and deploy a P2P platform **EApp** that achieves over 95% success rate in information exchange using low-cost unmodified Android phones.
- We demonstrate **EApp**'s applicability to emergency preparedness and response in disconnected rural areas and show that it can support information delivery within several hours.
- To account for the unique information needs and technological reliance of rural populations, we employ an iterative participatory design campaign with community partners and residents from a rural community in the US. Key insights are incorporated in the **EApp** design and implementation.
- We assess the feasibility of **EApp** to successfully deliver EPR information in rural areas through a combination of controlled in-lab experimentation and small-scale deployments. Our findings shed light on the energy and latency overhead of rural EPR information access through WiFi-Direct P2P networks.

2 RELATED WORK

Emergency-oriented applications strive to prepare the public for disasters [4, 17, 19, 22, 24, 25, 27] and support first responders [5, 29]. Many are customized to *a type of emergency*: medical [4], hazardous materials [22] and fire [19]; or to *the area they serve*: regional [24, 25] or national [17]. The majority of existing applications provide only static instructional content [4, 19, 22, 25]. Apps that also provide live contextual alerts [17, 24, 25, 27] require constant broadband access, as information is either pushed to the users [17, 24, 27], or users are referred to external websites [25]. In contrast, **EApp** aggregates all information types and enables information exchange among rural residents with intermittent Internet access via opportunistic P2P connectivity.

Offline chat platforms such as FireChat, Signal Offline, Serval and others [2, 6, 14, 26, 41] use mesh networking (typically over WiFi Direct), and are thus, conceptually similar to **EApp**. However, most of them do not detail the technical design, nor provide performance analysis. One exception is Serval [41], which was designed prior to battery optimizations introduced with newer Android versions [15] and its reliance on guaranteed synchronous background services makes it obsolete. In addition it employs WiFi and requires that peers act as both APs and clients to exchange information. In contrast, we enable P2P communication in the presence of Android battery optimization using native WiFi Direct and foreground services, and demonstrate its feasibility in both lab and real-world deployments.

Delay-tolerant (DTN) and opportunistic P2P networks have been explored for emergency response [30, 34, 42, 44, 46, 47, 50] with *primary focus on multi-hop routing* of messages between victims and first responders [30, 34, 43, 44, 47, 49]. [48] employs Bluetooth 5 technology to facilitate the routing of patient monitoring information to medical entities. For *physical connectivity*, some systems employ WiFi in ad-hoc mode on rooted (non-stock) phones [30], while others employ tethering to mimic infrastructure WiFi [44]. Authors of [30] analyze a realistic scripted disaster scenario to evaluate the efficiency of DTN routing protocols over ad-hoc WiFi. Our work is orthogonal as it focuses on the P2P data link layer as opposed to multi-hop routing. We tackle challenges of hands-free operation and global user synchronization in Wi-Fi Direct networks in stock Android. In addition, our work aims to disseminate emergency information to offline residents as opposed to connect victims with responders in offline settings.

WiFi Direct research focuses primarily on optimal group formation [35–37, 39, 40, 53, 54, 58]. [37] examines the time incurred for group formation as a function of the number of users attempting P2P connections and the WiFi Direct operation mode. [40] extends the legacy WiFi Direct operation to multi-group communication for multi-hop ad-hoc networking. Other work on group formation tackles network self-organizing and healing [39] by designing middleware abstractions [31] or introducing redundancy in P2P link formation for fast network recovery [36, 38]. Finally, [51] explores "group-less" WiFi Direct communication for establishing an emergency network at an emergency scene where devices can exchange messages. The majority of these works strive to maximize group size, while minimizing the time for group formation. A common assumption is that the group owner is known or selected ahead of time, which is not practical for opportunistic mobile networks. In addition, these approaches require modifications of the Android operating system, while we strive to deliver services on commodity devices. Our approach leverages the naturally-occurring group sizes. We show that most often the group size formed by stock Android devices is comprised of two phones. Thus, we extend group formation attempts based on the information exchange needs. Our work caters to the mobility of users while focusing on maximizing

[1]We develop and evaluate **EApp** on Motorolla G6 phones, which cost $150.

emergency information across the community and not restricted to a disaster area.

Battery life of smartphones in disaster scenarios has also been recognized as a major challenge [45]. Recent analyses characterize the power draw for link establishment over WiFi Direct [56], Wi-Opp [55] and Bluetooth. Since energy consumption varies for group owners and peers, a fair and adaptive role-switching is warranted [57]. While the above work focuses on scan and connect, we further quantify the effects of all stages of information exchange, including data transfer. The negative effects of Android power saving strategies on WiFi Direct's P2P connectivity have also been demonstrated and relevant predictive models for asynchronous communication proposed [32]. Our synchronous approach to circumventing these limitations ensures over 95% success rate in P2P exchanges. We believe there might be a potential for mixed synchronous (our work) and asynchronous [32] modes of operation for P2P interactions, which we plan to investigate in future work.

3 WIFI DIRECT PRELIMINARIES

We provide a brief review of WiFi Direct [33] as it is central to our system design. A client in a WiFi Direct network can be either a group owner (GO), acting as an access point in a star topology, or a peer. Before peers can exchange information, they establish a link, which involves: (i) device discovery and (ii) group formation.

Device discovery. To establish a group, peers discover each other by cycling through three phases: *scan*, *listen* and *search*. In the *scan* phase, a peer scans all supported WiFi channels for available WiFi Direct groups to join. In the absence of groups, the peer discovers other lone peers to form a group by switching between *listen* and *search* in the so-called *social channels* (1, 6 and 11 from the 2.4GHz range). To *search*, a peer sends Probe Requests for another peer to intercept and generate a corresponding Probe Response and trigger a group establishment procedure. If a searching peer does not receive a Probe Response within a predefined interval, it enters the *listen* phase, waiting to receive others' Probe Requests. To maximize the likelihood of rendezvous, peers randomize their length of time spent searching and listening.

Group formation. Once two devices have discovered each other, they proceed to form a group. Depending on how the GO is selected, we have *standard* and *autonomous* groups, and depending on the authentication requirements there are *persistent* and *non-persistent* groups. In standard groups the GO is negotiated upon each encounter, whereas in autonomous groups, a device unilaterally decides to be a GO. Persistent groups allow peers to reuse the roles and credentials established during their first encounter, while in non-persistent groups they are established anew at each encounter. While non-persistent groups require user's engagement to accept each connection request, persistent groups require this only once upon the first encounter of a peer. We use persistent group formation, thus handling variance in peer encounters with minimal user engagement.

4 DESIGN AND IMPLEMENTATION

In this section we discuss the **EApp** system design and implementation (Fig. 1), which is comprised of an Android application and a backend server. The server acts as an information aggregator

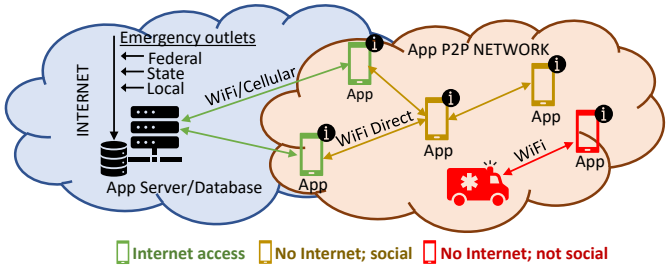


Figure 1: EApp system design.

Stage	Gr. #	Type	Participants	Date
1	1	Resp. (Firefighters)	10	3/1/19
	2	Resp. (Medical)	3	4/5/19
	3	Residents Cohort 1	8	3/15/19
	4	Residents Cohort 2	7	4/15/19
2	5	Res. Cohort 1 & 2	7	10/4/19
	6	Res. Cohort 1 & 2	2	11/15/19

Table 1: Focus groups

pulling from several information outlets identified by our community partners. Alerts and information are curated for offline users and stored into a database. The app is installed on community members' smartphones. We differentiate between two types of users: those with internet access and offline users. Users with Internet access can query the backend server for fresh information. They can further distribute the information to offline users in their community through P2P exchange over Wi-Fi Direct. As our goal is to maximize the utilization of emergency information by rural community residents, we co-designed both the user interface and the data aggregation in a sequence of IRB-approved focus group meetings with target users.

4.1 Participatory design of EApp's interface

To account for their unique information needs and promote rural users' technological reliance, our team conducted multiple on-site focus groups with residents and first responders in a rural community in New York (Table 1). In collaboration with the Director/Fire Coordinator at the county office of Emergency Services, we recruited two focus groups with first responders to gain their insights on information residents need in the wake of an emergency or in preparation for emergencies. We also recruited two cohorts of residents using a voluntary survey that was distributed at local public meetings and online. The survey collected information on demographics, smart phone and applications use, Internet access, travel and commuting habits, and interactions with other residents. The resident focus groups elucidated information needs and steered useful functions and features of **EApp**. The findings from this first stage were used to develop a first version of the user interface, which was then demonstrated to the community and further refined following a second co-design stage.

Next we summarize key findings from our co-design informing the user interface presented in Fig. 3.

(1) *User Settings and Preferences:* Participants desired to customize the types of pop-up notifications they would receive based on their location and visitors/resident status. They also wanted to customize features of the **EApp** interface such as the font size.
(2) *Alerts:* Participants requested multiple types of emergency alerts including weather conditions, road closures, hazardous materials, medical/health, fire, and missing persons. Multiple official sources were identified at different levels including FEMA and state and county agencies. Participants requested an emergency banner with real-time alerts, corresponding notifications, and the ability to track unfolding emergencies over time.
(3) *Preparedness Information:* Participants were also interested in preparedness information such as evacuation routes, first aid procedures, self-help guides, service facility locations (e.g. hospitals), and shelters.
(4) *News:* Participants wanted access to the latest local first responder news, including information from agencies' social media and websites.
(5) *Communication:* Participants suggested that the **EApp** should support two-way communication between local residents and first responders as well as between residents. The goal of resident-to-resident communication is to enable interactions and to support sharing of emergency or non-emergency information through photos, incidents or safety status. Residents-first responders communication would serve to notify first responders of emergency situations or to share personal profile information for residents experiencing an emergency. However, two-way communication between residents and first responders was deemed potentially disruptive by the latter group in the first two focus groups.

4.2 EApp Server

The server architecture is depicted on the right-hand side of Fig. 2. The core components of the server include (i) information scrapers, (ii) user management, (iii) and information access subsystem. The *information scrapers* execute every minute, querying IPAWS, NY511, NOAA, NYAlert and our collaborators' Facebook and Twitter pages. While the alert outlets serve self-contained short messages, the information sources often include a short message followed by a chain of web links, which one has to follow to obtain full information on a given issue. This is not possible for offline users and, thus, the **EApp** server curates information for offline use by crawling chains of URLs embedded in Twitter and Facebook messages. When information is curated, the original content is preserved and not modified in any way. The *user management* module handles account creation and password recovery. The *EPR information access* queries curated content from the database and sends it to the **EApp** clients over a secure HTTPS connection.

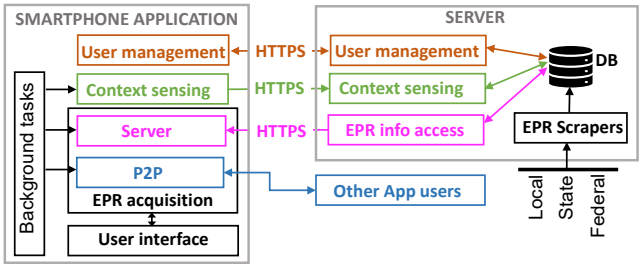
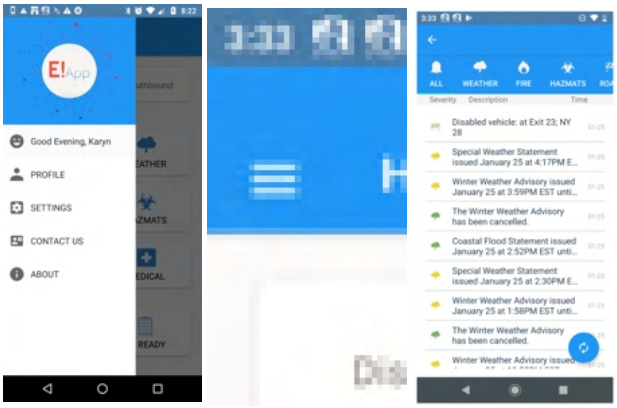


Figure 2: EApp architecture.



(a) Navigation **(b) Home** **(c) All alerts**

Figure 3: Snapshots of the EApp user interface.

4.3 EApp Android application

The **EApp** Android application (Fig. 2 left) is comprised of several modules including (i) a user interface (Fig. 3), (ii) EPR information acquisition, (iii) context sensing and (iv) user management. The context sensing and information acquisition modules subscribe to a background service, which ensures synchronous execution of the underlying networking services despite Android battery optimization.

The user management module allows users to create accounts and change their password. It also facilitates settings updates such as user type (visitor/resident), emergency contact information and notifications.

The context sensing module subscribes to the background service to periodically collect geo-tagged contextual information such as Bluetooth, cellular and WiFi network availability and signal strength. We employ the `WiFiManager API` [13] to scan available WiFi access points, `BluetoothAdapter API` [8] to scan available Bluetooth devices and `Telephony API` [11] to get information about cellular network services.

The information acquisition module includes two modalities for information access: (i) from the **EApp** server for users who have Internet access, and (ii) through a phone-to-phone exchange over WiFi Direct.We utilize the `HttpURLConnection API` [9] for online users to query the **EApp** server, and the `WiFiP2p API` [12] to establish P2P connections. Both client-server and P2P information exchanges begin with the exchange of vector clocks, which *eliminates the transmission of redundant information.* The timestamps in

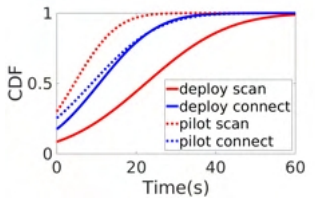


Figure 4: Scan and connect times in our deployments.

the vector clocks are compared and only the delta information is exchanged. The goal is to distribute all information while keeping the network traffic in check.

Privacy and authentication are also essential in **EApp**. All sensed context data is stored in an encrypted database on the phones. Sensitive data such as geolocations and passwords are encrypted at the server. For password recovery, we use three-way authentication that ensures password resets are not malicious. Finally, both the client-server and the P2P exchanges are encrypted. For client-server we use SSL encryption, while for P2P we employ WiFi Direct's WPA link-layer encryption. The **EApp** uses standard persistent groups for P2P exchange (§3). To ensure that connections are made only with legitimate users, we maintain and distribute a MAC address whitelist of all registered users.

The background service for periodic task execution is a key binding component that drives **EApp**'s information acquisition and context sensing. Maintaining seamless communication over socio-physical networks requires synchronous access to underlying networking services without explicit user engagement. Thus, our background service features three components:

(1) **Global clock.** In order to exchange information, two **EApp** users have to be in each other's vicinity and attempt to form a group at the same time. Thus, we implement a global clock and a corresponding global duty cycle (ΔT) to ensure that distributed **EApp** users perform synchronous peer discovery. This requires users' global clocks to be synchronized. To inform the necessary synchronization precision, we explore the WiFi Direct peer discovery behavior over hundreds of encounters from our pilot and rural deployments (§5.4). Fig. 4 presents discovery (scan) and connect times. The median scan time for the pilot is 5 seconds, whereas in the deployment it is 23 seconds. This is due to the larger link distances in the deployment, triggering phones to scan longer before connecting. The median time to form a P2P group (connect) is 10 seconds both in the pilot and in the rural deployment. Thus, as long as phones are synchronized within the observable median scan times, they can discover each other.

(2) **Persistent network access.** Automatic information exchange requires periodic access to the device networking services without user intervention. This brings forth a classical trade-off in mobile systems between service availability and battery life. To balance this trade-off, Doze Mode [15] was introduced in the Android API v.23 (2015) and refined ever since. The inherent principle of Doze Mode is that all access of background processes to networking/location services is halted unless the user is actively engaging with their phone.

To circumvent this issue, we experimented with several options to keep the phone out of Doze Mode and ensure seamless information exchange. We began by employing a background service and Alarm Manager [7] to periodically execute a WiFi Direct scan. In addition, we white listed **EApp** to be exempt from battery optimization [15]. This combination, however, failed to keep the phone out of Doze Mode and the resulting exchange success rate was a mere 20%. Simply exempting an app from battery optimization does not ensure persistent access to networking services. Wakelock [10] offers an alternative by keeping the display dimmed, thus, preventing the phone from entering Doze Mode. While this approach enables exchange success rates of nearly 100%, it leads to an infeasible battery overhead.

Ultimately, we modified our approach in a way that was minimally disruptive to the user, while allowing high and consistent success rate of information exchange. To this end, we implement **EApp** as a foreground service and exempt the **EApp** from battery optimization. All periodic processes including WiFi Direct information exchange and context sensing subscribe to the foreground service, which is unaffected by Doze Mode. In addition, we changed the alarm type that triggers scan execution to *setAlarmClock*, which wakes the device up prior to executing a scan. The combination of these implementation choices allowed the **EApp** to operate on schedule resulting in over 90% exchange success rate without significant battery overhead (more details in §5). While the foreground service is a great tool to override the Doze Mode, a mobile OS may stop the service if it detects unexpected battery consumption. The **EApp** is designed with a mechanism to restart the foreground service should the OS interfere with the service.

(3) **Maximizing information exchange in rural socio-physical networks using Wi-Fi Direct.** When in each other's vicinity, peers need to maximize the information exchange in order to ensure rapid propagation of emergency information across the community. To inform our maximal information exchange approach, we perform an empirical evaluation of the WiFi Direct group formation dynamics. According to the specification, if a set of peers D are collocated, they can form a group of size $|D|$ and exchange information [28]. In reality, stock WiFi Direct does not always maximize the group size [37]. Our evaluation (§5.3) demonstrates this via longitudinal controlled experiments with increasing peer set sizes (2, 3, and 4). Our 4-phone experiments demonstrate that stock WiFi Direct is only able to form a group of 4 phones in 1 out of a 100 attempts, with the majority of attempts resulting in 2-peer groups. Thus, a single-shot approach, whereby each peer attempts a single discovery/connection per duty cycle results in severe deterioration of the success rate for information exchange.

To ensure maximal information exchange, we develop a multi-shot peer discovery and information exchange protocol, which is illustrated in Fig. 5. The protocol employs a duty-cycling approach, whereby the beginning of the global clock t_0 is set to midnight of the current day. A fixed duty

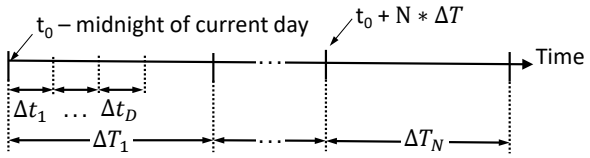


Figure 5: Overview of the EApp multi-shot peer discovery and information exchange protocol.

Algorithm 1: **EApp** multi-shot peer discovery and information exchange protocol.

```
1:  INPUT W, {P, D, C} = ∅
2:  BEGIN Duty Cycle; Mini Cycle
3:  P=Find-EApp-Peers();
4:  if P !empty then
5:      D = P ∩ W − C {Check if devices in whitelist}
6:      if D !empty then
7:          G=FormGroup(D) {Returns users in group}
8:          for i =0; i < |G|; i++ do
9:              S_i=ExchangeWith(D_i)
10:             if S_i is success then
11:                 C = C + Di
12:                 D = D − Di
13:             end if
14:         end for
15:     else
16:         END Mini-Cycle {Done exchanging; fall idle}
17:     end if
18: else if P empty then
19:     END Mini-Cycle {No peers found; fall idle}
20: end if
```

cycle, ΔT, periodically repeats for as long as the app is running. Within each duty cycle, the app executes a number of mini-cycles Δt_d, $d \in (1, |D|)$, where D is the set of **EApp** peers encountered in the given duty cycle. Each mini-cycle caters to a single peer discovery attempt, whereas a sequence of mini-cycles implements our multi-shot approach.

Alg. 1 describes the **EApp** multi-shot peer discovery and information exchange protocol. The algorithm executes at the beginning of each duty cycle. It starts by performing a peer discovery (line 3), which returns a list of collocated WiFi Direct peers P (**EApp** and non-**EApp**). P is then checked against the **EApp** peer whitelist W (line 5). If **EApp** peers were discovered, the protocol will attempt to form a group (line 7) and exchange information (line 9). Upon successful exchange with a peer D_i, that peer is removed from the set D and added to the set of peers C with whom the device has exchanged (lines 11 and 12). The mini-cycle repeats until Δt_D, in which all discovered peers have been communicated with (i.e. the list D is empty). The app will then fall into idle state until the next duty cycle.

Our multi-shot procedure ensures that all collocated peers D participate in information exchange regardless of whether

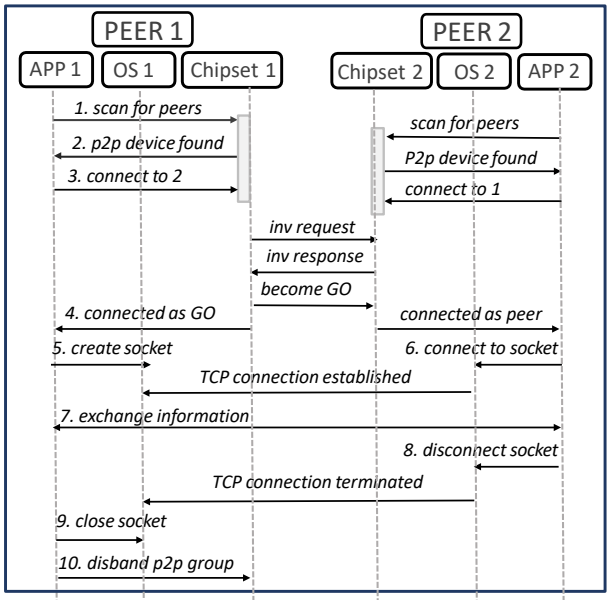


Figure 6: OS calls and message exchanges for an encounter between two EApp peers (Peer 1 is GO and Peer 2 is Peer). Numbered steps designate logging points for our evaluation.

a group of size $|D|$ is formed in the first mini-cycle or not. In terms of complexity:

- *A minimum of one mini-cycle* occurs when a device attempts a scan but no **EApp** peers are discovered. The device will remain idle for that duty cycle (line 19).
- *Two mini-cycles* occur if the WiFi Direct protocol succeeds in forming a group that includes all discovered peers D at once. The algorithm will execute one more mini-cycle to check for new peers before falling idle.
- *A maximum of D mini-cycles* may occur if the WiFi Direct protocol executes its worst-case group formation of 2-peer groups regardless of the number of discovered peers D. In this case, the protocol will execute $|D| - 1$ mini-cycles for information exchange and one additional mini-cycle to ensure that there are no remaining peers.

Our protocol balances the tradeoff between energy saving and information exchange maximization in rural socio-physical networks whilst operating on off-the-shelf Android. In cases where peers are not available, the protocol saves energy by falling into idle mode. When WiFi Direct is efficiently able to form a maximal-sized group, the protocol harnesses this opportunity to exchange with all peers. Finally, if WiFi Direct forms groups inefficiently, our protocol still ensures maximal information exchange for the added cost of multiple peer discovery attempts. This cost is justified by the opportunity to relay fresh emergency information.

5 EVALUATION

In this section we evaluate the **EApp**, focusing on its ability to support successful information exchange in rural disconnected areas. This success hinges on several factors including the robustness and persistence of establishing P2P links, operational distances, battery

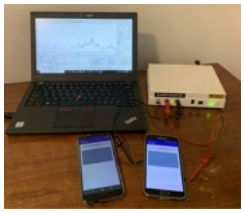


Figure 7: Monsoon setup

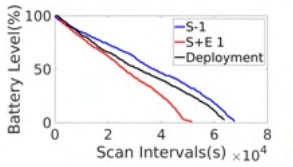


Figure 8: Battery discharge

overhead and information exchange capabilities. We show that the **EApp** facilitates successful information exchange while harnessing casual user interactions.

5.1 Experimental Methodology

We begin by describing our measurement methodology. Fig. 6 presents an illustration of operating system (OS) calls and the corresponding message exchanges for one **EApp** peer encounter. Actions in each peer are broken down based on where they are initiated and where they execute (i.e. the App, the OS or the WiFi Chipset). The app logs events at each of the numbered steps (1-10). As detailed in Alg. 1, an encounter begins with a call from the App through the OS `WiFiP2p API` to the Chipset requesting a scan for peers (step 1). In step 2, the Chipset returns a set of peers found. The App then requests a connection with specific peers (step 3). Following this request, the Chipsets handle the group negotiation and WiFi Direct authentication, while the OS handles the IP address assignment for the group. The App is then informed by the Chipset that a group was formed and which devices are in the group (step 4). The App then proceeds to establish the corresponding TCP sockets for information exchange, logging when the socket is established (step 5), when information sending starts (step 7), when does it end and how much data was exchanged (steps 8 and 9). Finally, the phone logs the time at which the WiFi Direct group is disbanded (step 10). Logs are recorded in the **EApp**'s SQLite database and offloaded to the server for analysis.

Metrics. Using the logs, we calculate several metrics to evaluate the **EApp**'s performance. *Group formation success rate* is defined as the fraction of duty cycles in which a peer is able to form a group. *Information exchange success rate* is the fraction of groups in which a peer was able to send and receive information with other peers. We also analyze the information *delivery time*, defined as the time from information birth to its delivery to a users. *Latency* and *throughput* are calculated at the TCP transport layer, and capture P2P communication performance once a group is formed.

Experimental setups. We use three setups:
Fine-grained power measurements. We evaluate the power consumption of the **EApp**, as this will be a primary driving factor for its usability. To this end, we use the setup depicted in Fig. 7 , which consists of a Monsoon power meter [18] connected to a Samsung Galaxy S5 Duos phone. We bypassed the phone's battery, connecting the phone to the main channel of the power meter. This allowed us to both power up the phone and measure its energy consumption. We sampled power draw (in mW) at a fine temporal granularity of 200μs, allowing us to profile the power consumption of various processes initiated by the **EApp**, including peer discovery, group formation and information exchange.

Controlled app analysis. We used a set of Moto G6 phones with the **EApp** to evaluate performance in controlled indoor and outdoor settings. We varied group sizes, distances between peers and the amount of data exchanged and assessed the success rate in peer discovery and information exchange, throughput and latency.
Deployments. We undertook two IRB-approved deployments: (i) a semi-staged pilot with seven users, and (ii) an ongoing deployment with residents in our partner community upstate New York (details in §5.4).

5.2 EApp power consumption

We begin by evaluating **EApp**'s battery discharge rate in controlled and deployment settings. Fig. 8 presents our results, where S-1 (blue) is the discharge rate for peer discovery, while S+E 1 (red) is the discharge rate for peer discovery and information exchange in a controlled experiment. The black curve presents discharge rate during our rural deployment. For discovery and exchange, the battery discharges in approximately 50,000 seconds (or 14 hours), whereas with peer discovery only, the battery lasts for over 19 hours. The discharge rate from our deployment falls between the S-1 and S+E 1, since not all duty cycles resulted in actual exchanges. These results indicate that even with persistent information exchanges and a short duty cycle, the phone is able to sustain its operation with a single charge per day (assuming the phone is plugged in overnight). Of note is that the battery drain profiles from Fig. 8 were taken on the Group Owner (GO), which inherently incurs a larger power drain than the Peer [57]. Thus, we anticipate that the battery life of **EApp** users who serve as Peers will be even longer.

Next we focus on the power draw for peer discovery (i.e. scan), group establishment (i.e. connect) and information exchange across distances with Line-of-Sight (LoS) and non-Line-of-Sight (nLoS) between two phones. For this experiment, we used our Monsoon setup, measuring current draw. Fig. 9 shows the floor plan of our department, where the experiments were performed. Fig. 10 shows our results. 10a shows the current draw over time for a single run with an annotation of the segments corresponding to peer discovery, group formation and exchange. Fig. 10b and 10c present scatterplots of current draw across five runs with increasing distances and LoS/nLoS, respectively. First, we note that the peer discovery (scan) cost is negligible compared to group formation and exchange. Second, the cost does not seem to increase significantly with distance. This might change at larger distances (i.e. approaching 600ft), as rate adaptation might select lower data rates, leading to longer data transmission delays and corresponding increases in power draw. Finally, nLoS links obstructed by multiple obstacles significantly impact the group formation and exchange capabilities, whereby exchanges fail after 50ft (3 walls separation) and group formation fails after 75 feet (5 walls separation). These results are conservative, as each distance increment adds one more wall (solid concrete or metal stud framing) between the WiFi Direct peers.

5.3 Maximizing information exchange

In this section we evaluate **EApp**'s information exchange capabilities. We assess the throughput and latency as a function of data size and link distances. We also analyze the success rate in P2P group formation and P2P data exchange.

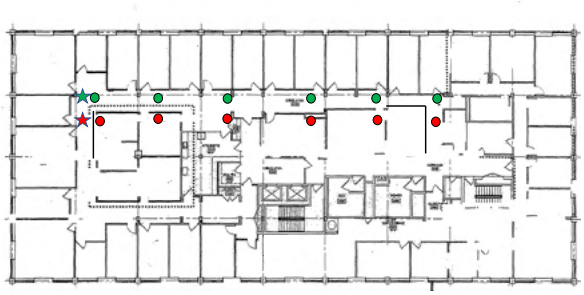
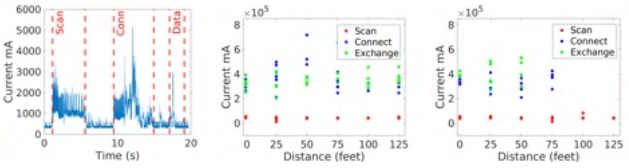


Figure 9: Floor plan for power draw experiments. Star designates the peer attached to the Monsoon power meter. Dots designate the other peer in LoS (green) and nLoS (red).



(a) Scan profile **(b) LoS** **(c) nLoS**

Figure 10: Current draw evaluation. 10a – current draw over time for one attempt of scan, connect and exchange. 10b and 10c – current draw with distance (LoS/nLoS).

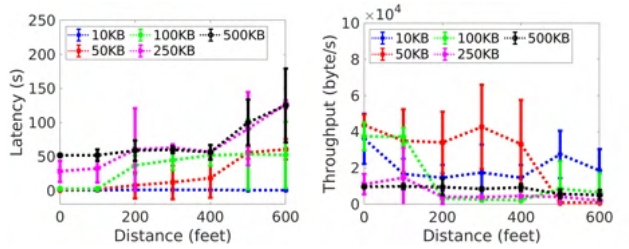


Figure 11: Latency (left) and throughput (right) across distances and transmission data sizes.

5.3.1 Throughput. We set up two Moto G6 phones with the **EApp** and have them transmit an increasing amount of data ({10KB, 50KB, 100KB, 250KB, 500KB}) using TCP. The choice of TCP over UDP for this experiment was purposeful, as the **EApp** information exchange also uses TCP for reliability. We move one of the phones away from the other in increments of 100ft from 0 up to 600ft (the maximum supported distance by WiFi Direct). At each distance/data size setting we perform 10 runs. Fig. 11 presents our results. The latency (left) remains stable for small data sizes (10KBytes) and grows with distance as the data size increases. The larger the data size, the higher the latency with distance. Naturally, we see the reverse trend with throughput (right), whereby the highest throughput is achieved with small data sizes and the throughput decreases as the data size grows. This can be explained in part with TCP's congestion window control, which directly affects the rate at which data is sent. At longer distances links are likely experiencing channel variations, which, in turn, affect the latency and the corresponding TCP sender rate. Overall, these results indicate that the necessary peer encounter duration has to be between 1 and 50 seconds for a successful exchange. In addition, the results underline the importance of limiting the amount of exchanged data to 10-100KBytes (e.g. by pruning stale updates) to ensure rapid exchange. These results are commensurate with our deployment findings (§5.4).

5.3.2 Group formation statistics. We now analyze **EApp**'s ability to form groups in a controlled setup. We compare two counterparts of the **EApp** P2P protocol: single-shot and multi-shot. Single-shot allows only one group attempt per duty cycle, whereas multi-shot is our exhaustive approach described in Alg. 1, which continuously attempts peer connections until all peers in the vicinity have been contacted or the duty cycle runs out.

For two phones over 100 duty cycles, the success rate is 98% with both protocols. While this is not surprising, it demonstrates the success of our implementation efforts to ensure persistent network access (§4.3).

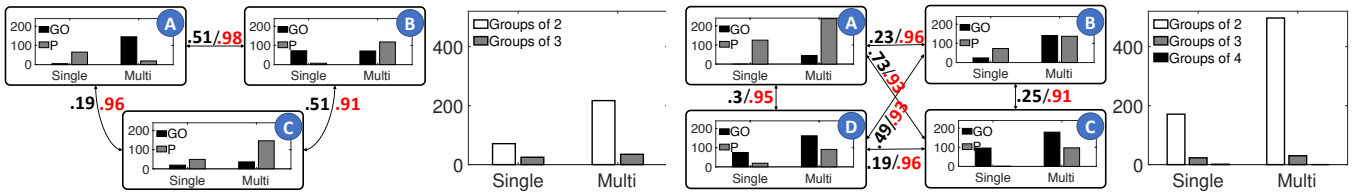
The benefits of our multi-shot approach are truly emphasized when the user population grows beyond two phones. Fig. 12 presents results from two experiments with three (12a,12b) and four phones (12c,12d) over 100 duty cycles. The pairwise phone interactions are represented as graphs (12a,12c). Each node represents one phone. Statistics inside the nodes represent the amount of encounters in the single- and multi-shot protocol in which that phone was a Group Owner (GO) or Peer (P). Edges are drawn between phones that interacted in a group. Edge labels present the pairwise exchange success rate, defined as the fraction of the total 100 duty cycles in which the corresponding phones were in a group. Black labels represent single-shot, whereas red labels represent multi-shot success rate. The single-shot approach significantly impacts a device's ability to exchange information, as the average success rate is 0.4/0.37 in the three- and four-phone setup, respectively. Some pairs were particularly affected by low success rate, e.g. A-C in the three-phone setup and A-B, A-D, B-C and C-D in the four-phone setup, as their ability to form a group was hampered by others' preference to each other. The multi-shot approach improves the average success rate to 0.95/0.92 for the three- and four-phone setup, and maintains a success rate of 0.91 or higher for all communicating pairs. This obviates unfairness in information access across peers and ensures high success in data exchanges.

Finally, in Fig. 12b and 12d, we examine achieved group sizes. The vast majority of groups were of size 2 even though we had three or four collocated phones. Out of a 100 duty cycles, the phones were able to form a group of three in 23/35 of the attempts and a group of four in a single instance. These findings further emphasize the importance of our multi-shot P2P protocol.

5.4 EApp performance in real deployments

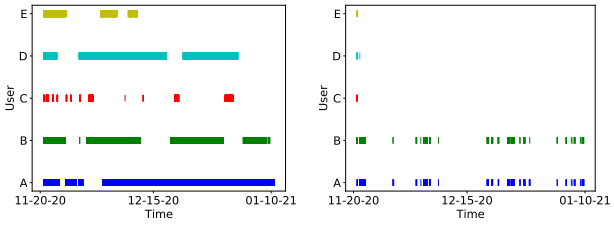
We evaluated **EApp**'s applicability to real-world settings in two IRB-approved deployments: (i) *a semi-staged pilot* and (ii) *an in-situ deployment* in a rural community upstate New York. For both deployments we used low-cost Moto G6 phones running stock AndroidOS version 9. Our deployments allowed us to evaluate incident delivery time, in-situ success rate of peer discovery and exchange, throughput and latency.

5.4.1 Deployment details. **Semi-staged pilot.** Our semi-staged pilot included seven participants over the course of five days in



(a) Exchange success rate 3 phones (b) Group size 3 phones (c) Exchange success rate 4 phones (d) Group size 4 phones

Figure 12: Group sizes and pairwise exchange success rate of the single- and multi-shot protocol with three (12a,12b) and four (12c,12d) collocated phones. 12a and 12c are graph representations of phone interactions. Each node represents one phone. Edges are drawn between phones that were in a group. Statistics within the node indicate the amount of group formations in which that phone was a group owner (GO) or a Peer (P). Edge labels indicate the pairwise exchange success rate of single- (black) and multi-shot (red). Single-shot has a low average success rate and is unfair to some pairs. Multi-shot brings the average success rate to 0.94 and ensures high exchange success rate for each possible pair. 12b and 12d show that despite a larger number of collocated devices, the native WiFi Direct is most successful in forming groups of size 2.



(a) User activity (b) User encounters

Figure 13: User behavior during the rural deployment.

August 2020. Four of the users were couples who lived together. The remaining three users lived in three different locations. Users were mostly going about their daily business, except some of them met as a group once a day. Five of the phones were offline and two phones had Internet access (one through residential WiFi and the other through a mobile subscription). Three of the five residences are in proximity within the maximum WiFi Direct range. Throughout the pilot, we injected new incidents on the server every ten minutes.

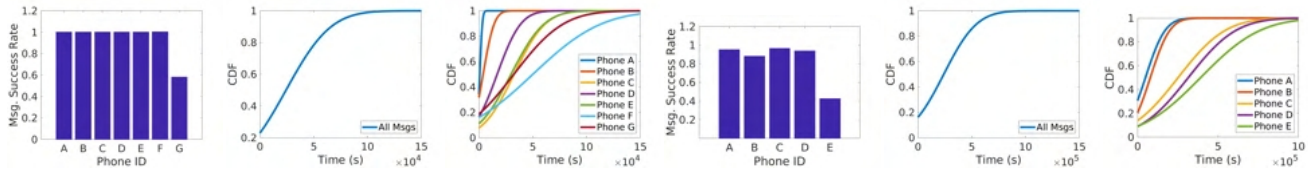
Rural deployment. The **EApp** is planned for a multi-stage deployment to fifty users. Our first stage began on November 20th, 2020 and is still ongoing, and includes five users. The findings reported in this paper span the period November 20th, 2020 - January 10th, 2021. All users reside in separate dwellings however, two of the users are in close proximity (within the maximum WiFi Direct range). Four of the users have internet access at home while the fifth user lives in a remote location without Internet access. Unlike the pilot, incidents distributed to our users are solely those incoming from official sources. Fig. 13a presents the periods during which users kept their phones *ON*, while Fig. 13b indicates user encounters over time. A majority of the encounters occurred between users A and B, who live in the neighboring residences. User encounters happen in 4.3% of the entire deployment period.

EApp bootstrapping. **EApp** uses standard persistent groups for P2P exchange §3. To ensure phone interactions, we undertook a bootstrapping process to form all possible groups before we distribute the phones.

5.4.2 Peer discovery and information exchange. We begin by evaluating the success rate of peer discovery and information exchange during our pilot. Table 2 lists for each of the seven users, the actual scans (1), the number of instances in which peers were found (2), the number of established connections (3) and the number of information exchanges (4). The table also lists in bold the *connection success rate* (the ratio between the number of connections and the number of instances peers were found) and the *exchange success rate* (the ratio between the number of information exchanges and the number of established connections) for each peer. These results grant several important observations. First, considering the 576 expected scans and the actual scans (1), we see that the app is able to consistently and reliably access the underlying networking resources. Considering the number of scans (1) and the number of instances in which peers were found (2) we see that in a large fraction of the cases, peers were not found. This is expected, since peers were only collocated in limited amount of the time. Second, considering the connection success rate, we see that with some exceptions, in 70-80% of their encounters peers were able to connect. The lower success rate of Peers B and E (both based in Residence 2) can be explained in part with the relative proximity of Residence 1 and Residence 2, which are neighboring households at about 100 feet distance. Due to this proximity, we observed multiple occasions, in which a peer from the pool {A, B, D, E} was found, but the link was not stable enough (due to distance and obstructing walls) for a connection to be established. In addition, Residence 2 has a much larger square footage than Residence 1. This created additional challenges in link establishment between peers B and E (collocated in Residence 2), which were not present for peers

Events/Phone	A	B	C	D	E	F	G
(1) Actual Scans	548	576	576	576	576	494	377
(2) Inst. Peers Found	359	278	52	345	216	162	101
(3) Connections	284	140	38	275	85	117	79
(4) Info. Exchanges	280	139	36	272	84	111	74
Connection success rate	**0.79**	**0.50**	**0.73**	**0.79**	**0.39**	**0.72**	**0.78**
Exchange success rate	**0.98**	**0.99**	**0.95**	**0.98**	**0.99**	**0.95**	**0.94**

Table 2: Success rate of peer discovery and data exchange by device during our pilot over five days (576 duty cycles).



(a) Success rate (b) Delivery time (c) Deliv. time/user (d) Success rate (e) Delivery time (f) Deliv. time/user

Figure 14: Evaluation of message distribution efficiency during our pilot and rural deployment. (14a, 14d) message delivery success rate across users; (14b, 14e) message delivery time for all users; (14c, 14f) message delivery time across users. EApp achieves nearly 100% message delivery rate with a median delay of 7 hours for the pilot and 65 hours for the rural deployment.

A and D (collocated in Residence 1). Despite the fact that **EApp**'s connection success rate was lower than a 100%, the app was able to deliver messages to all users in a timely manner (within a few hours in the worst case; details below). Finally, considering the exchange success rate, we see that as long as the peers were able to connect, a successful exchange followed in nearly 100% of the cases.

We also examine the group formation dynamics with WiFi Direct in uncontrolled settings. Although there were ample opportunities for groups larger than two peers, a majority of the groups during the pilot (98.8%) were of size 2 and the remaining 1.2% were of size 3. At the same time, our logs show that there were 103 duty cycles (or about 18% of all deployment duty cycles), in which more than two phones were collocated. This further highlights the importance of our multi-shot approach in maximizing the information exchange.

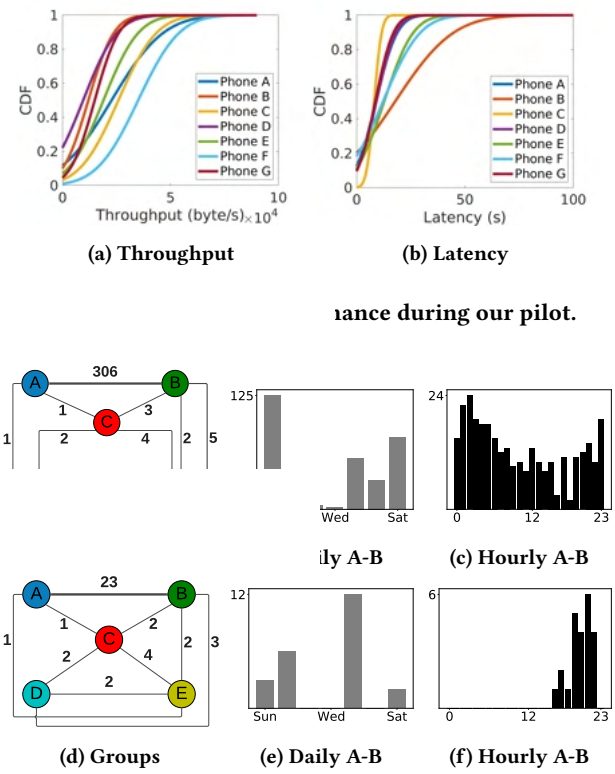
5.4.3 Timeliness of information dissemination. Next, we evaluate the completeness and timeliness of information delivery to **EApp** users in our deployments. We begin by evaluating the percentage of all messages that were successfully delivered to users in the pilot (Fig. 14a) and the deployment (Fig. 14d). A total of 605 messages were sent during the pilot and 141 during the deployment. Six of the seven pilot users received a 100% of these messages. User G (offline, not social) stopped interacting with the group two days into the deployment, and as a result, their message delivery rate was 58%. Similarly, four out of five deployment users received nearly all messages. User E (mostly offline, not social) received 40% of the messages (Fig. 13).

We also assess the timeliness of information dissemination by analyzing the message delivery time defined as the difference between message delivery on a given phone and the time at which the message was generated on the server. Fig. 14b and 14e present aggregate delivery time across all users for the pilot and deployment, respectively. 25% of the pilot and 20% of the deployment messages were delivered instantaneously. The median delivery time was 7 hours for the pilot and 65 hours for the deployment, whereas the worst case delivery was 41/416 hours. Breaking down these statistics by user (Fig. 14c, 14f) we see that the timeliness of message delivery varied substantially across users, depending on their Internet access and social behavior. In the pilot, user A (blue), which had cellular access, received all messages within 10 minutes (i.e. a single duty cycle). User B (orange), who had residential WiFi access, received messages nearly instantaneously in 35% of the cases and not slower than 4 hours in 90% of the cases. The remainder of the users, who were all offline (C-G), experienced a variety of information delivery delays, depending on their social behavior. D, who was collocated with A (the phone with cellular access) for a large portion of the day, was the next most successful to receive messages with a median delay of about 4.5 hours. Next, E, who was collocated with B (the phone with WiFi access) experienced a median delay in the order of 8 hours. The remaining users (C, F and G) experienced a median of 9, 9 and 13 hours, respectively and a long tail of up to a day for C, E and G and 41 hours for F. Similarly in the deployment, users who had residential WiFi access (A and B) experienced a median delay of 13 hours, whereas these with less-frequent access had a median delay of 72 hours.

Overall, these findings bring a few important insights, which while not necessarily surprising, highlight that the **EApp** is able to successfully leverage users' socio-physical networks for timely dissemination of emergency information. First, even for users who are not particularly social, information can be delivered within a few hours if they live in neighboring households. Second, offline users who spend more time with Internet users incur a lower delay in message delivery. Third, even if users are isolated for most of the time, a brief daily interaction (e.g. by a target dispatch) can significantly improve their access to emergency information.

5.4.4 Latency and throughput. We evaluate the latency and throughput for all exchanges during the pilot. Fig. 15 presents these across devices. The median achieved throughput is between 10 and 35kbps, whereas the 95th percentile is between 28 and 61kbps. The median latency is between 8 and 18 seconds with 95th percentile between 13 and 53 seconds. Phone B, which had residential WiFi access experienced the longest delays in P2P information exchange. We postulate this might be due to contention of both WiFi Direct and WiFi services on the same chipset. Of note is that the **EApp** uses TCP for reliability, thus, these reported values incorporate TCP rate adaptation. In addition, we note that both the latency and throughput results from our deployment are commensurate with these from our controlled experiments, presented in Fig. 11. These values inform the necessary encounter duration for successful information exchange.

5.4.5 Community deployment user interactions. We analyze the social encounters of users in our deployment and the corresponding P2P exchanges those encounters facilitated. We represent the encounters and P2P group formations as social graphs in Fig. 16a and 16d, respectively. The majority of encounters and corresponding group formations occurred between users A and B, who live within 100 feet of each other. Although these users had 306 encounters, they were able to form a group only 23 times. To further



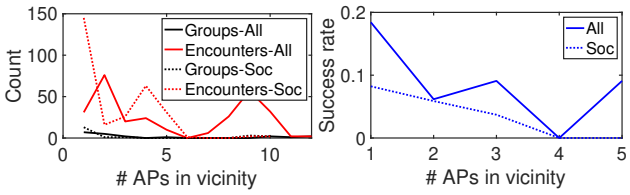
(a) Throughput (b) Latency

ıance during our pilot.

(c) Hourly A-B

(d) Groups (e) Daily A-B (f) Hourly A-B

Figure 16: Analysis of user encounters (16a, 16b, 16c) and group formations (16d, 16e, 16f) in our rural deployment. The majority of encounters and groups occurred between users A and B. We break down A-B's interactions over days of the week and hours of the day in 16b, 16c, 16e and 16f. While encounters occur uniformly, a majority of the groups f



(a) Encounters/groups (b) Success rate

Figure 17: Influence of external WiFi interference on group formation.

explore these interactions, we analyze encounters and group formations over the days of the week (Fig. 16b, 16e) and hours of the day (Fig. 16c, 16f). While encounters occurred in all days of the week and hours of the day, group formations were predominantly successful in the evening (6PM-10PM).

There are several factors that can hamper group formation. First, since the same WiFi chipset is used for both P2P and infrastructured WiFi, contention on the chipset might cause lower P2P success rate. In addition, external interference from infrastructured WiFi, which by design emits at higher power levels than WiFi Direct, can also be a factor, especially when infrastructured WiFi operates on channels 1, 6 and 11 (the "social" channels used for peer discovery in WiFi Direct). Fig. 17 seeks to quantify the effects of external interference on the A-B group formations. Fig. 17a presents the number of encounters (red) and group formations (black) with increasing number of infrastructured access points (AP) in the peers' vicinity. Solid line presents all encounters/groups with any AP in vicinity, whereas dashed line presents all encounters/groups with APs on the social channels. Considering the group formations (black lines), we see that the fewer APs in vicinity, the more likely it is that a group would form. We further expand this analysis in Fig. 17b, which presents the success rate in group formation (the ratio of formed groups vs. all encounters) considering all APs in the peers' vicinity (solid) and APs on social channels only (dashed). The success rate decreases as the number of collocated APs grows. Furthermore, the success rate with APs on social channels is consistently lower than that with any APs. Of note is that these results are based on limited data and as our deployment continues they will become more conclusive. Despite this, we already see that the success rate of WiFi Direct P2P networks depends on the presence and use of infrastructured WiFi. This grants a rethink in peer discovery to aid infrastructureless communications.

6 DISCUSSION AND FUTURE WORK

This paper builds and evaluates the **EApp** framework for information exchange in disconnected rural areas. Its operational distances and communication modalities make the **EApp** applicable across various scenarios from preparing users for how to act in the face of an emergency to supporting information dissemination in actual emergencies. While we make first steps towards the system realization, there are several directions that will further improve **EApp**'s applicability.

Battery optimization. While the current version of the **EApp** already allows the phone to operate with one charge per day, there are several avenues to further improve the battery life of **EApp** peers. First, we can optimize the peer discovery intensity by fine-tuning the duty cycle from a set of possible values that are multiples of each other. The intensity can be informed by the situation (e.g. an emergency will trigger high intensity) or the likelihood of peer encounter. We can also leverage the predictability of users socio-physical interactions to optimize peer discovery and save battery.

Adaptive data transmissions. Our results indicate that the transmitted data size has an impact on the latency and, in turn, the necessary encounter duration. We can use these insights to fine-tune the volume of exchanged data in order to maximize the reach of information.

Larger rural deployment. To fully assess **EApp**'s potential to deliver information in rural areas, we are planning a larger deployment with our community partners. The design of the **EApp** allows for seamless integration of new users into the existing deployment

7 CONCLUSION

While our society critically relies on rural areas for its sustenance, rural emergency preparedness and response significantly lag behind their urban counterparts. A key barrier is the lack of reliable broadband, which hampers residents' ability to receive emergency

information from their agencies. To develop self-reliance, rural communities often create tight-knit social structures and take charge of their own technological progress.

Our efforts strive to leverage these social structures to improve the access to emergency information in rural areas. To this end, we develop the **EApp**: an Android application that aggregates information from various agencies at the federal, state and local level and serves it to rural offline users through opportunistic P2P exchange. We developed the **EApp** using an iterative co-design process that engaged rural first responders and residents from a partner community. We evaluated **EApp**'s performance through extensive lab experimentation and small-scale deployments, and made several important observations that speak to the **EApp**'s applicability to rural scenarios. First, we showed that the app does not incur prohibitive battery use; there are multiple avenues still open to further minimize the impact on power consumption. Second, we demonstrated that the typical latency for information exchange is 8-18 seconds, which can be realistically accommodated by users natural interactions. Finally, in small-scale deployments, we showed that the typical delay for message delivery is in the order of several hours, even when users follow fairly independent daily routines. The **EApp** is developed for low-cost stock Android devices with an outlook towards wide applicability. The **EApp** is compatible across Android devices and Android devices with different OS's because we are leveraging stock Android.

ACKNOWLEDGMENTS

This work is supported by NSF S&CC grant CMMI-1831547 and NSF CAREER grant CNS-1845858.

REFERENCES

[1] 2018. Federal Communications Commission 2018 Broadband Deployment Report. https://www.fcc.gov/reports-research/reports/broadband-progress-reports/2018-broadband-deployment-report.
[2] 2019. FireChat. https://www.cbc.ca/news/technology/firechat-off-the-grid-messaging-app-what-you-need-to-know-1.2784271.
[3] 2019. Rural Health Information Hub. https://www.ruralhealthinfo.org/topics/emergency-preparedness-and-response-requirements.
[4] 2020. First Aid: American Red Cross. https://apps.apple.com/us/app/first-aid-american-red-cross/id529160691.
[5] 2020. IAMRESPONDING: Emergency Responder Reply System. https://iamresponding.com/v3/Pages/Default.aspx.
[6] 2020. Vojer. https://apps.apple.com/us/app/vojer-be-connected-conference-or-in-roaming-be-intouch/id913585553.
[7] 2021. Android AlarmManager. https://developer.android.com/reference/android/app/AlarmManager.
[8] 2021. Android Bluetooth. https://developer.android.com/guide/topics/connectivity/bluetooth.
[9] 2021. Android Http Connection. https://developer.android.com/reference/java/net/HttpURLConnection.
[10] 2021. Android PowerManager. https://developer.android.com/reference/android/os/PowerManager.
[11] 2021. Android Telephony. https://developer.android.com/reference/android/telephony/package-summary.
[12] 2021. Android WiFi Direct (Peer_to_Peer). https://developer.android.com/guide/topics/connectivity/wifip2p.
[13] 2021. Android WiFiManager. https://developer.android.com/reference/android/net/wifi/package-summary.
[14] 2021. BRIAR. https://briarproject.org/index.html.
[15] 2021. Doze Mode in Android. https://developer.android.com/training/monitoring-device-state/doze-standby.
[16] 2021. Federal Emergency Management Agency: Integrated Public Alert and Warning System. https://www.fema.gov/integrated-public-alert-warning-system.
[17] 2021. FEMA Smartphone App. https://www.fema.gov/mobile-app.
[18] 2021. Monsoon Solutions.
[19] 2021. National Fire Protection Association Mobile Apps. https://www.nfpa.org/Codes-and-Standards/All-Codes-and-Standards/NFPA-digital-products.
[20] 2021. New York Alert. https://alert.ny.gov/.
[21] 2021. New York State Department of Transportation: 511NY Traffic alerts. https://511ny.org/.
[22] 2021. NIOSH Pocket Guide to Chemical Hazards Mobile Application. https://www.cdc.gov/niosh/npg/mobilepocketguide.html.
[23] 2021. NOAH: National Weather Service. https://www.weather.gov/.
[24] 2021. Notify NYC: NYC Emergency Management. https://www1.nyc.gov/site/em/resources/notify-nyc-app.page.
[25] 2021. Rensselaer County Bureau of Public Safety app. https://play.google.com/store/apps/details?id=com.ocv.rensselaercountynyem&hl=en_US.
[26] 2021. Signal offline messenger. https://play.google.com/store/apps/details?id=com.raxis.signalapp.
[27] 2021. The MDSS Mobile App. https://www.dtn.com/the-mdss-mobile-app/.
[28] 2021. WiFi Alliance. https://www.wi-fi.org/discover-wi-fi/specifications.
[29] 2021. WISER. https://wiser.nlm.nih.gov/.
[30] Flor Álvarez, Lars Almon, Patrick Lieser, Tobias Meuser, Yannick Dylla, Björn Richerzhagen, Matthias Hollick, and Ralf Steinmetz. 2018. Conducting a large-scale field test of a smartphone-based communication network for emergency response. In *Proceedings of the 13th Workshop on Challenged Networks*. 3–10.
[31] Luciano Baresi, Naser Derakhshan, Sam Guinea, and Francesco Arenella. 2017. Magnet: A middleware for the proximal interaction of devices based on wi-fi direct. In *2017 IEEE International Conference on Communications (ICC)*. IEEE, 1–7.
[32] Elisabetta Biondi, Chiara Boldrini, Andrea Passarella, and Marco Conti. 2017. What you lose when you snooze: How duty cycling impacts on the contact process in opportunistic networks. *ACM Transactions on Modeling and Performance Evaluation of Computing Systems (TOMPECS)* 2, 4 (2017), 1–29.
[33] Daniel Camps-Mur, Andres Garcia-Saavedra, and Pablo Serrano. 2013. Device-to-device communications with Wi-Fi Direct: Overview and experimentation. *IEEE wireless Communications* 20, 3 (2013), 96–104.
[34] Claudio Casetti, Carla Fabiana Chiasserini, Luciano Curto Pelle, Carolina Del Valle, Yufeng Duan, and Paolo Giaccone. 2015. Content-centric routing in wi-fi direct multi-group networks. In *2015 IEEE 16th international symposium on a world of wireless, mobile and multimedia networks (WoWMoM)*. IEEE, 1–9.
[35] Claudio Ettore Casetti, Carla Fabiana Chiasserini, Yufeng Duan, Paolo Giaccone, and Andres Perez Manriquez. 2017. Data connectivity and smart group formation in Wi-Fi direct multi-group networks. *IEEE transactions on network and service management* 15, 1 (2017), 245–259.
[36] Wael Cherif, Muhammad Asif Khan, Fethi Filali, Sanaa Sharafeddine, and Zaher Dawy. 2017. P2p group formation enhancement for opportunistic networks with wi-fi direct. In *2017 IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE, 1–6.
[37] Marco Conti, Franca Delmastro, Giovanni Minutiello, and Roberta Paris. 2013. Experimenting opportunistic networks with WiFi Direct. In *2013 IFIP Wireless Days (WD)*. IEEE, 1–6.
[38] Utku Demir, Aaron Faulkenberry, Cristiano Tapparello, and Wendi Heinzelman. 2018. Reducing delay in group reformation in WiFi direct networks through redundancy. In *2018 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 1–7.
[39] Utku Demir, Cristiano Tapparello, and Wendi Heinzelman. 2017. Maintaining connectivity in ad hoc networks through wifi direct. In *2017 IEEE 14th International Conference on Mobile Ad Hoc and Sensor Systems (MASS)*. IEEE, 308–312.
[40] Colin Funai, Cristiano Tapparello, and Wendi Heinzelman. 2017. Enabling multi-hop ad hoc networks through WiFi Direct multi-group networking. In *2017 International Conference on Computing, Networking and Communications (ICNC)*. IEEE, 491–497.
[41] Paul Gardner-Stephen, Romana Challans, Jeremy Lakeman, Andrew Bettison, Dione Gardner-Stephen, and Matthew Lloyd. 2013. The serval mesh: A platform for resilient communications in disaster & crisis. In *2013 IEEE Global Humanitarian Technology Conference (GHTC)*. IEEE, 162–166.
[42] Stephen M George, Wei Zhou, Harshavardhan Chenji, Myounggyu Won, Yong Oh Lee, Andria Pazarloglou, Radu Stoleru, and Prabir Barooah. 2010. DistressNet: a wireless ad hoc and sensor network architecture for situation management in disaster response. *IEEE Communications Magazine* 48, 3 (2010), 128–136.
[43] Pan Hui, Jon Crowcroft, and Eiko Yoneki. 2008. BUBBLE Rap: Social-based Forwarding in Delay Tolerant Networks. In *MOBIHOC 2008*. Hong Kong SAR, China.
[44] Andre Ippisch and Kalman Graffi. 2017. Infrastructure mode based opportunistic networks on android devices. In *2017 IEEE 31st International Conference on Advanced Information Networking and Applications (AINA)*. IEEE, 454–461.
[45] Watcharachai Kongsiriwattana and Paul Gardner-Stephen. 2016. Smart-phone battery-life short-fall in disaster response: Quantifying the gap. In *2016 IEEE Global Humanitarian Technology Conference (GHTC)*. IEEE, 220–225.
[46] Zongqing Lu, Guohong Cao, and Thomas La Porta. 2017. Teamphone: Networking smartphones for disaster recovery. *IEEE Transactions on Mobile Computing* 16, 12 (2017), 3554–3567.
[47] Abraham MartíN-Campillo, Jon Crowcroft, Eiko Yoneki, and Ramon Martí. 2013. Evaluating opportunistic networks in disaster scenarios. *Journal of Network and*

computer applications 36, 2 (2013), 870–880.

[48] Esther Max-Onakpoya, Oluwashina Madamori, Faren Grant, Robin Vanderpool, Ming-Yuan Chih, David K. Ahern, Eliah Aronoff-Spencer, and Corey E. Baker. 2020. Augmenting Cloud Connectivity with Opportunistic Network for Rural Remote Patient Monitoring. In *International Conference on Computing, Networking and Communications (ICNC): Wireless Ad hoc and Sensor Networks*.

[49] Hrushikesh Mehendale, Ashwin Paranjpe, and Santosh Vempala. 2011. LifeNet: A Flexible Ad Hoc Networking Solution for Transient Environments. In *SIGCOMM 2011*. Toronto, Ontario, Canada.

[50] Christian Meurisch, The An Binh Nguyen, Stefan Wullkotte, Stefan Niemczyk, Florian Kohnhäuser, and Max Mühlhäuser. 2017. NICER911: Ad-hoc Communication and Emergency Services Using Networking Smartphones and Wireless Home Routers. In *Proceedings of the 18th ACM International Symposium on Mobile Ad Hoc Networking and Computing*. 1–2.

[51] Meng-Shiuan Pan and Chung-Ming Wang. 2019. A group-less and energy efficient communication scheme based on Wi-Fi direct technology for emergency scenes. *IEEE Access* 7 (2019), 31840–31853.

[52] Andrew J Prelog and Lee M Miller. 2013. Perceptions of disaster risk and vulnerability in rural Texas. *Journal of Rural Social Sciences* 28, 3 (2013), 1.

[53] Ahmed A Shahin and Mohamed Younis. 2014. A framework for P2P networking of smart devices using Wi-Fi direct. In *2014 IEEE 25th Annual International Symposium on Personal, Indoor, and Mobile Radio Communication (PIMRC)*. IEEE, 2082–2087.

[54] Ahmed A Shahin and Mohamed Younis. 2015. Efficient multi-group formation and communication protocol for wi-fi direct. In *2015 IEEE 40th Conference on Local Computer Networks (LCN)*. IEEE, 233–236.

[55] Sacha Trifunovic, Maciej Kurant, Karin Anna Hummel, and Franck Legendre. 2015. WLAN-Opp: Ad-hoc-less opportunistic networking on smartphones. *Ad Hoc Networks* 25 (2015), 346–358.

[56] Sacha Trifunovic, Andreea Picu, Theus Hossmann, and Karin Anna Hummel. 2013. Slicing the battery pie: Fair and efficient energy usage in device-to-device communication via role switching. In *Proceedings of the 8th ACM MobiCom workshop on Challenged networks*. 31–36.

[57] Sacha Trifunovic, Andreea Picu, Theus Hossmann, and Karin Anna Hummel. 2014. Adaptive role switching for fair and efficient battery usage in device-to-device communication. *ACM SIGMOBILE Mobile Computing and Communications Review* 18, 1 (2014), 25–36.

[58] Hongxu Zhang, Yufeng Wang, and Chiu C Tan. 2014. WD2: An improved WiFi-direct group formation protocol. In *Proceedings of the 9th ACM MobiCom workshop on Challenged networks*. 55–60.