How Do I Get This App? A Discourse on Distributing Mobile Applications Despite Disrupted Infrastructure

Amro Al-Akkad
Fraunhofer Institute for
Applied Information Technology (FIT)
Sankt Augustin, Germany
amro.al-akkad@fit.fraunhofer.de

Christian Raffelsberger
Institute of Information Technology/Lakeside
Labs, Alpen-Adria-Universität Klagenfurt
Klagenfurt, Austria
christian.raffelsberger@aau.at

ABSTRACT

This paper aims to lead a discourse on distributing mobile applications while having no access to cloud services. While in daily life people acquire applications via application stores, the access to those stores can be severely hampered in disasters. Instead of wishful thinking, i.e. hoping that people or manufactures would pre-install an (emergency) app before a disaster, we have started to investigate into Internet-less deployment mechanisms. We discuss five potential approaches of deploying apps in an ad-hoc fashion. Each approach is assessed against six criteria, while taking a stance that the smartphone is the minimally required deployment platform. This paper concludes with the observation that current mobile operating system providers do not provide "Internet-less" deployment mechanisms, although platforms as Android allow for this. This said, we hope that this contribution can spark further interest into the discussed problematic.

Keywords

Deployment, Smartphones, Disrupted Infrastructures, Software Distribution Platforms

INTRODUCTION

In the course of expanding connectivity due to broadband coverage (UMTS, LTE) or public Wi-Fi hotspots and consequently an increasing proliferation of powerful portable devices (smartphones, tablet computers), people have adopted to use, in daily life and in disasters, mobile applications, shortly apps, that give access to a range of information and communication technology (ICT) services. ICT services include texting services such as SMS or WhatsApp, map-based services such as GoogleMaps, social networks such as Facebook, or microblogging such as Sina Weibo or Twitter. However, in the aftermath of disasters those ICT services can be disturbed severely due to disruptions of the underlying network infrastructure. Such disruptions and outages may be a result of damaged or overloaded network infrastructure. Currently, most everyday solutions such as Facebook/Twitter clients or specific emergency apps, such as the FEMA App[1] or NowForce[2], run on top of pre-existing network infrastructure and thus are subject to become unhelpful for communicating people's needs if the infrastructure is disrupted

In the course of recent disaster events, two phenomena can be observed. On the one hand, people reconfigure common apps to become useful despite disrupted infrastructure. On the other hand, people and authorities promote apps that provide means to communicate despite disrupted infrastructures. Regarding the latter, for

[1] FEMA App http://1.usa.gov/P8V5sf

[2] NowForce Inc. http://www.nowforce.com/

instance, Al-Akkad et al. (2013) report that in the aftermath of the 2013 Boston Bombings people and authorities tweeted to share OpenGarden[3], an app which facilitates interconnectivity between proximate mobile devices via Bluetooth. With regard to the former phenomenon, Ukrainian and Venezuelans protesters[4] used an app called Zello, which enables users to send short voice messages from person to person or to groups. As Zello, has been blocked by a specific Venezuelan network provider, the app developer revised the code for changing IP addresses, which makes it more difficult to block the app.

Some recent research efforts have investigated similar ad-hoc communication approaches that require no pre-existing infrastructure. For example, Twimight (Hossmann et al. 2011) provides a Twitter client that enables users to exchange tweets between neighboring devices via Bluetooth or common Wi-Fi, until one device may be able to upload them to the Internet. The Help Beacons system (Al-Akkad et al. 2014) enables trapped or buried casualties to broadcast a distress signal via the Wi-Fi interface of their smartphone, which in turn first responders can discover, and track with a seeker app. However, many of those approaches result in dedicated apps, which people would rather not download before a disaster from a distribution store or website.

This paper tries to fill the gap between the need for (disaster) applications that work despite disrupted network infrastructures and the current deployment channels that require access to the Internet (e.g., common application stores). Thus, instead of relying on smartphone providers to install such dedicated apps upfront on the underlying operating system, this paper looks into approaches that facilitate the ad-hoc deployment of apps. Currently, we are at an early stage of design and thus we would like to use this paper as a sort of discussion platform. Since apps and the corresponding app stores recently emerged, we are not aware of any related work of deploying apps in an ad-hoc fashion. Of course, various works exist on distributing data via some protocol. For example, the work of Hall et al. (2009) describes the Domino platform enabling mobile phone users to share recommendations between neighboring mobile devices. However approaches like Domino require at least one minimal module of functionality being already deployed. In the following we present the core idea of each approach we have considered so far. Beforehand, we introduce the frame of our project and six criteria to assess the advantages and disadvantages of each approach.

BACKGROUND AND CRITERIA FOR DEPLOYMENT ON "AIR-GAPPED" WIRELESS DEVICES

The frame of our research is a European project that aims to improve interoperability–technical and social–in the context of large-scale crisis management. In this context, we explore several ideas to support ad-hoc networking in different scenarios and for different device configurations, from specific wireless router devices to common smartphones. While we explored and demonstrated approaches towards mobile applications that can be useful despite disruptions of the pre-existing network infrastructure, often questions regarding the deployment arose. In this context, it is obvious that most people would not download a specific emergency app from a distribution store beforehand and application stores may not be available during a disaster situation. One solution would be that phone vendors or mobile operators ship their smart phones with pre-installed disaster apps or integrated mechanisms to share apps offline one day. However, as it is probably a lengthy and difficult task to convince vendors to change the current deployment mechanisms, this paper focuses on off-line deployment approaches that work without vendor support.

As a result of these ongoing discussions we can think of five different deployment approaches and also selected six different criteria that we use to classify these approaches. While doing so, we take a stance viewing the smartphone itself as the minimal deployment platform. The classification criteria are as follows:

Hardware Requirements. This criterion describes which hardware is required to facilitate an approach. Approaches that only rely on smartphones have the lowest requirements.

Readiness. This criterion defines when it will likely be possible to establish a deployment system in the aftermath of a disaster, which of course strongly depends on the availability of the required hardware (e.g. the readiness decreases if additional hardware needs to be deployed first)

Familiarity. This criterion expresses how familiar people are with a specific technique. The familiarity is high if people know similar mechanisms beforehand. For example, people may already use their smartphone to scan QR codes for URLs in their daily life.

Practical feasibility. This criterion denotes to which extent a deployment approach can be implemented on a smartphone. For example, some approaches may only require off-the-shelf smartphones, whereas others need

[3] OpenGarden Inc., https://opengarden.com/

[4] CNN Global News View, http://edition.cnn.com/2014/02/24/world/venezuela-ukraine-protests-apps/

specially prepared (e.g a rooted Android phone, a jailbreaked iPhone) phones or a combination of smartphones and additional devices, such as routers or cell towers.

User Involvement. This criterion evaluates to which extent a user needs to be involved during obtaining an app. The user involvement is low if the deployment process is automated to a large extend. For example, scanning a QR code that triggers to access a web page is an example for high user involvement.

Ease of use. This criterion intends to assess the complexity of the process that users need to go through for obtaining an app. For example, typing a URL in a browser is rather complex compared with scanning a QR code. Scanning a QR has a high user involvement but the ease of use is low as it is not complex.

"INTERNET-LESS" APPROACHES FOR DEPLOYING MOBILE APPLICATIONS

This chapter illustrates five approaches for distributing apps to "air gapped" phones, i.e. phones that have no access to the Internet to access application stores.

Download via Ad-hoc Networks and Captive Portal Deployed on Mesh Nodes

The first approach is based on the presence of an ad-hoc deployed wireless network (e.g. Wi-Fi routers or cell towers). In previous disasters such as the 2009 L'Aquilla earthquake in Italy, some mobile operators extended their coverage with additional mobile stations[5] in order to cover homeless camps. This observation shows that people or organizations may respond by the deployment of network devices filling gaps of communication. Similarly, an open (i.e. unencrypted) Wi-Fi network could be deployed and configured that would provide means to distribute an emergency app. Figure 1a shows how users could access this network and as soon as they try to access the Internet, would be automatically redirected to a special web page that could include information how to acquire the app (e.g. provide a download link). The web page would be hosted locally on the deployed device and hence be available despite the lack of Internet access. This deployment approach is based on the captive portals technique that people normally experience after connecting to public Wi-Fi hotspots deployed at universities or hotels. From a deployment perspective the main disadvantage of this approach is that fire fighters or other responders need to be able to practically deploy or carry wireless devices for providing such captive portals. Furthermore, as this approach cannot be implemented on off-the-shelf smartphones, it is not possible that the app is re-shared among users. Instead, every user has to connect to the ad-hoc network and download the application.

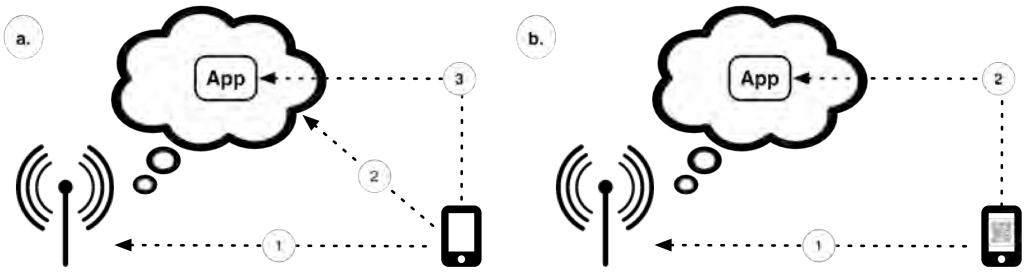

Figure 1. a. Captive Portal: 1) Connect 2) Redirect 3) Download
b. Captive Portal with QR Code: 1) Connect 2) Scan QR code triggering Download

Download via Ad-hoc Networks with QR Code

The second approach extends the previous approach by leveraging on the increasing proliferation of QR codes. When deploying the ad-hoc network in the aftermath of the disaster, leaflets or posters including a QR code that advertise an emergency app could also be distributed. Figure 1b illustrates such an example. Assuming that a device is already connected to a local Wi-Fi network, scanning a QR code with the phone's camera would result in being directed to a static page offering the download link for the emergency app. Besides deploying an ad-hoc device, this approach also requires to place QR codes on walls or other places inside the affected disaster zone.

[5] MEDC Earthquake Case Study - L'Aquila, Italy, Online Geography Resources, http://bit.ly/15pi20b

Cell Broadcast

This approach is similar to the previous two, as it requires additional infrastructure. However, this approach assumes that some pre-existing infrastructure, namely cell towers as depicted in Figure 2, is still operational. Cellular networks provide communication services that allow one-to-many communication. For instance, the GSM standard defines Cell Broadcast (CB) which is a mobile service to send short text messages with a length of up to 93 characters to all users in a network cell. Similarly, UMTS offers Multimedia Broadcast Multicast Services (MBMS) that allows cell towers to transmit multimedia data (e.g. mobile TV and radio broadcasting) and files.

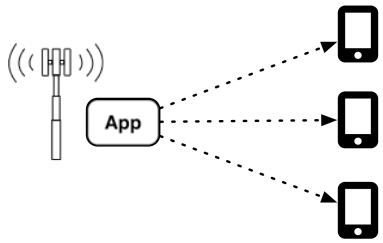

Figure 2. Cell Broadcast: Send messages to all phones residing in a cell

If a cell tower survives the adverse effects of a disaster, this approach is a probably the fastest way for distributing apps to people carrying "air gapped" phones. Currently, people are rather unfamiliar with this service as it recently emerged and is not widely deployed by mobile providers. Thus, the main downfall of this approach is that it requires the support of mobile providers that have to trigger the deployment of the emergency application. Additionally, mobile devices may not support or have disabled the reception of cell broadcasts.

Viral Deployment

Compared to all other approaches the approach, illustrated in Figure 3a, requires no infrastructure at all. It follows the work presented in (Al-Akkad et al. 2014) to exploit the SSID (i.e. the name of a Wi-Fi network) to broadcast a need in a certain range. In our case this would be to include a URL within the Wi-Fi SSID that points to a web page distributing the emergency app. The URL has to be static, e.g. "GOTOURL192.168.43.1:9999" and point to a web page that provides a download link for the application. In order to acquire the application, users would need to connect to an open (i.e. unencrypted) Wi-Fi network and type in the advertised URL which points to a dedicated site for distributing the app which is hosted on the device that provides the Wi-Fi hotspot. This approach can be implemented on off-the-shelf Android smartphones and hence also supports the re-distribution of the emergency apps between users. For instance, once the emergency app has been installed, it could open another Wi-Fi network advertising the download link and hence be distributed from phone to phone. While this approach is the only "infra-less" one, it is at the same time the least user-friendly one. Its use is rather limited for technology-savvy people, since common people may not understand the message conveyed in the SSIDs.

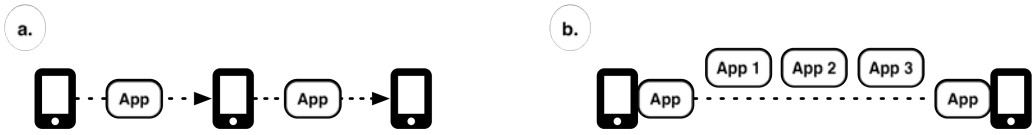

Figure 3. a. Viral Deployment (infra-less) b. Viral Deployment via App

Viral Deployment via App

To ease the style of user interaction described in the last approach, we propose the use of a generic app that can handle the exchange of other apps (see Figure 3b). Such a generic distribution app could offer a catalogue of apps that can be exchanged between smart phones without requiring Internet access. Besides disaster situations, any other contexts, in which spontaneous deployment of apps would be desired, could be addressed. For example, recently some airlines started to offer passengers to consume multimedia services via 'on board' available Wi-Fi hotspots. However, in order to join the network passengers have to pre-install a dedicated app from the airline. This presents a weakness or constraint in situations where consumers are not aware of such an application. This approach would be easier to use than the previous one, as the user has to interact with one application only. Preferably, the exchange of apps would be handled over a standardized and open protocol.

SUMMARY AND CONCLUSION

In this paper, we have presented five approaches to distribute apps among "air gapped" phones. Table 1 contrasts the different approaches against the above-defined six criteria. Currently, there exists not much research about sharing apps independent of cloud services. Though, some work is evolving in similar research projects to ours. For instance, the Serval project[6] also looks into ad-hoc communication and the dynamic deployment of mobile applications via captive portals[7]. In future, we will observe such concurrent approaches. Another aspect of future work is to observe works in the hacking community. A community being well known for creative approaches finding "trap doors" in current protocols and standards (Conti 2005). While in everyday life such trap doors might present a menace, at the same time in the light of disaster situations with disrupted network infrastructure they might be useful; for example, in order to distribute applications between neighboring devices. Of course, user permissions should precede this dissemination of data or apps.

Criteria / Approach	Captive Portal	QR Code	Cell Broadcast	Viral	Viral via App
Hardware requirements	~	–	–	+	+
Readiness	~	–	~	+	+
Familiarity	+	~	~	–	–
Practical Feasibility	–	–	+	+	~
User Involvement	+	+	+	–	~
Ease of Use	+	~	+	–	+

Table 1. Comparison of different deployment approaches. ("+" indicates the criteria is a strength, "–" indicates the critera is a weakness and "~" indicates the criteria is neither a strength nor a weakness)

This paper concludes with the finding that current available technology in smartphones does not really foresee to download applications from neighboring devices including ad-hoc deployed Wi-Fi hotspots or pass them from phone to phone. However, such distribution mechanisms may be crucial to distribute emergency applications. This said, this paper closes with the appeal to rethink or revisit mechanisms to deploy apps independent from the use of app stores.

ACKNOWLEDGEMENTS

The presented work is partially supported by the BRIDGE project. BRIDGE is a collaborative project co-funded by the EU Seventh Framework Programme (FP7SEC-2010-1) under grant agreement n°261817.

REFERENCES

1. Al-Akkad, A., Ramirez, L., Boden, A., Randall, Dave, and A. Zimmermann. (2014) Help Beacons: Design and Evaluation of an Ad-Hoc Lightweight S.O.S. System for Smartphones. *In Proceedings of the 32nd International Conference on Human Factors in Computing Systems* (*accepted for publication*).
2. Al-Akkad, A., Ramirez, L., Denef, S., Boden, A., Wood, L., Büscher, M., and A. Zimmermann (2013). "Reconstructing Normality": The Use of Infrastructure Leftovers in Crisis Situations As Inspiration for the Design of Resilient Technology. *In Proceedings of the 25th Australian Computer-Human Interaction Conference: Augmentation, Application, Innovation, Collaboration*, 457–466.
3. Conti, G. (2005). Why Computer Scientists Should Attend Hacker Conferences. *Communications of the ACM*, 48, 3, 23–24.
4. Hall, M., Bell, M., Morrison, A., Reeves, S., Sherwood, S. and M. Chalmers (2009). Adapting Ubicomp Software and Its Evaluation. *In Proceedings of the 1st ACM SIGCHI Symposium on Engineering Interactive Computing Systems*, 143–148.
5. Hossmann, T., Legendre, F., Carta, P., Gunningberg, P., and C. Rohner. (2011) Twitter in disaster mode: Opportunistic Communication and Distribution of Sensor Data in Emergencies. *In Proceedings of the 3rd International Conference on Communication and Computing*, 1–6.

[6] Serval Project, http://www.servalproject.org/

[7] Serval Mesh Extender, http://servalpaul.blogspot.co.at/2013/11/dont-panic-its-serval-mesh-extender.html