Social-aware Device-to-Device Communication: A Contribution for Edge and Fog Computing?

Huber Flores
University of Oulu, Finland
hflores@ee.oulu.fi

Rajesh Sharma
University of Bologna, Italy
rajesh.sharma@unibo.it

Denzil Ferreira
Chu Luo
Vassilis Kostakos
University of Oulu, Finland
{firstname.lastname}ee.oulu.fi

Sasu Tarkoma
University of Helsinki, Finland
sasu.tarkoma@helsinki.fi

Pan Hui
HKUST, Hong Kong, China
panhui@cse.ust.hk

Yong Li
Tsinghua University, China
liyong07@tsinghua.edu.cn

UbiComp/ISWC '16 Adjunct, September 12-16, 2016, Heidelberg, Germany.

DOI: http://dx.doi.org/10.1145/2968219.2968589

Abstract
The exploitation of the opportunistic infrastructure via Device-to-Device (D2D) communication is a critical component towards the adoption of new paradigms such as edge and fog computing. While a lot of work has demonstrated the great potential of D2D communication, it is still unclear whether the benefits of the D2D approach can really be leveraged in practice. In this paper, we develop a software sensor, namely *Detector*, which senses the infrastructure in proximity of a mobile user. We analyze and evaluate D2D on the wild, i.e., not in simulations. We found that in a realistic environment, a mobile is always co-located in proximity to at least one other mobile device throughout the day. This suggests that a device can schedule tasks processing in coordination with other devices, potentially more powerful, instead of handling the processing of the tasks by itself.

Author Keywords
Mobile Offloading; Internet-of-Things IoT; Context-aware; Mobile Cloud

ACM Classification Keywords
C.1.3 [Computer Systems Organization]: Other Architecture Styles—*Cellular architecture (e.g., mobile)*; C.2.4 [Computer-Communication Networks]: Distributed Systems—*Client/Server*

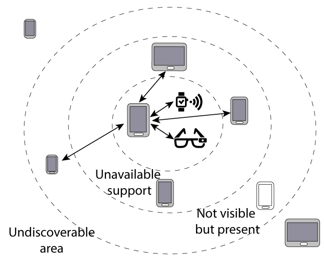

Figure 1: D2D discovery in proximity

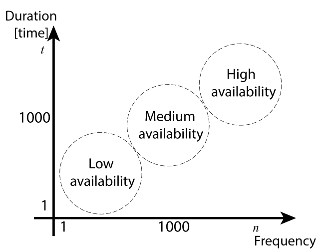

Figure 2: Qualitative stability of D2D

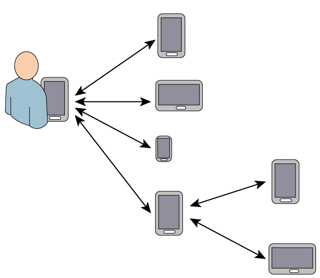

Figure 3: D2D coordination using a hierarchical dissemination approach

Introduction

Device-to-Device (D2D) Communication [5] is an important technology for the next generation of mobile network infrastructures. It aims to augment the constrained capabilities of a mobile device, to enhance the Quality-of-Service (QoS) and Quality-of-Experience (QoE) of the end user. Simply put, D2D is the opportunistic junction of devices that forms a temporal and resourceful computation infrastructure, which can be used to facilitate the completion of a task by distributing the complexity of its execution across multiple devices.

D2D exploits the opportunistic context of the devices, e.g., same location, several communication channels, higher data bandwidth and speed, etc, in order to create a more powerful infrastructure. The effort of executing a task using such infrastructure is reduced when compared with the effort required by a single device to execute the same task. For instance, a big file divided into smaller chunks can be uploaded or downloaded faster by multiple devices rather than a single device streaming the complete file. In addition, a computational task, e.g., QuickSort, NQueens, can be assigned to multiple devices to perform load-balancing, reducing energy consumption.

Naturally, since the formation of a D2D infrastructure is not controlled by a centralized authority, e.g., network operator, but rather it depends on the mobility and interest of users to lease their mobile devices' resources, a D2D infrastructure relies on the egocentric behavior of the user. Therefore, until now most of the work is focus on measuring the potential conceptual benefits of D2D environments and investigating mechanisms to foster user's participation.

While a lot of work has demonstrated the potential of D2D communication, it remains unclear the benefits of D2D approach in practice. D2D is a critical component towards the adoption of new paradigms such as *Edge Computing* and *Fog Computing*, in which mobile devices are utilized to support the processing of others devices [3], e.g., sensors. In this work, we focus on answering: *how much D2D can deliver on the wild?*

We develop a software sensor called *Detector*, which collects information about the devices located in proximity of the user. By analyzing the traces of 10 participants (collected during one month), we find that, on average, a mobile user is able to encounter at least one device throughout the day. This result suggests that a device schedule coordinated task processes with other devices, instead of handling them single-handed.

Related Work

Network performance — Multiple D2D approaches for packet forwarding and data offloading in wireless networks aim to release the network from excessive data traffic, and to accelerate the end-to-end transmission of data for applications [14, 10, 15].

Transient infrastructure — Middlewares to coordinate mobile devices has been proposed in order to facilitate the creation of opportunistic infrastructure [2, 1, 12], which can be used to balance the processing workload of the mobile applications via computational offloading [9, 7]. This also includes the utilization of D2D as edge devices for Internet of Things (IoT) and offloading support devices for fog environments [3, 8].

Social participation — Since the participation of the mobile users is a key factor for the adoption of D2D, multiple works have analyzed the conceptual gains which can be exploited from human mobility [11]. On the other hand, a lot of work oriented to improve the social participation have been also proposed [2, 13].

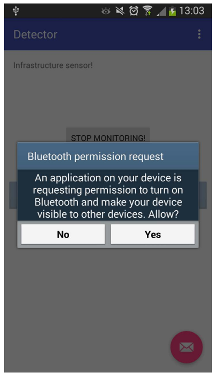

Figure 4: *Detector* application requesting explicit permission for Bluetooth activation to the user

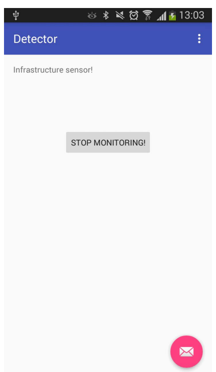

Figure 5: *Detector* application to be located in the background.

D2D Infrastructure Sensor

In fog and edge computing contexts, it is expected that a D2D infrastructure will process a computational task from other devices in proximity, e.g., sensors, smart-watches, smart-glasses, etc. While the transfer of a computational task among devices is a trivial task in low latency networks, the formation of a D2D infrastructure is a complex task as it requires, 1) the devices to be discoverable among them (Figure 1), 2) the establishment of stable intercommunication channels (Figure 2), and 3) the coordination of the devices in order to distribute the processing load (Figure 3).

Naturally, a device that is not discoverable, it is not addressable by others. Thus, it cannot be exploited for the creation of opportunistic infrastructure. A device can be addressable by others via Bluetooth or WiFi-Direct technologies. However, the addressability of a device depends on the mobile user, which is the one that grants visibility and permissions to others. In this work, we focus on the discoverable issue, and left the rest as future work.

In order to measure how much D2D can contribute to fog and edge computing environments, we develop the *Detector* application, searches for available devices by relying on Bluetooth and WiFi-Direct. The discovery of devices happens each 15 minutes. We consider this interval to be enough in order to avoid disturbing the charging behavior of mobile user. By default, Bluetooth and WiFi-Direct induces high energy drain the device. Thus, if the discovery process is trigger to often, then it is likely that the user needs to charge his/her devices at different hours than the regular ones.

The developed application stores traces about the devices found in a SQLite database, which contains information in the key-value format, which includes the type of device, the name of the device, battery discharge rate, Bluetooth and WiFi-Direct addresses. The *Detector* application is hosted in Google app store (beta service in order to facilitate its deployment and distribution. The logic of the application is quite simple for the user, who just have to grant permission to the application to use the Bluetooth of his/her device (Figure 4). Once the permission is granted, the application can collect information in the background, which means that *Detector* does not interfere with the usage of the mobile device (Figure 5). In order to achieve this, the application implements a scheduling routine that activates the discovery process of each network interface. Each day the applications uploads the database to a remote server located in Amazon EC2.

Evaluation and Analysis

In this section, we quantify the amount of infrastructure that can be found in proximity of the mobile user.

Setup and methodology: — The goal of this experiment is to identify nearby infrastructure which can be sensed by a user's smartphone on a regular basis based on his/her mobility. We recruit 15 participants to install the *Detector* application. We then collect daily information of each participant during one month experiment. From the 15 participants, we discard 5 as the collected data from those users did not contain enough information to re-construct the sensing process during each day.

Results: — Based on the data collected, we calculate the total number of mobile devices that were discovered by each participant. Figure 6 shows the results for WiFi-Direct and Bluetooth. While some participants were able to discover a large number of devices (p0, p1, p6, p7, and p9), some others discovered a few devices (p2, p3, p4 and p8). This is reasonable as different users have different daily life's routines. Also, some users are more active than oth-

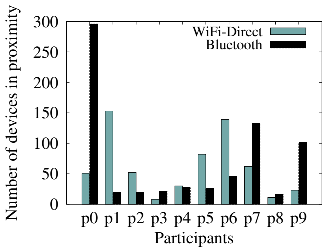

Figure 6: Number of devices detected per each participant that installed the *Detector application*

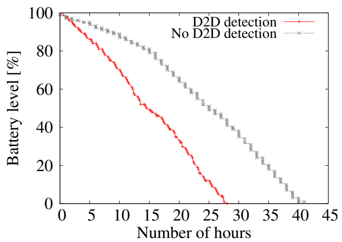

Figure 7: Energetic effort (battery life) required for a particular device to detect available devices in proximity using *Detector*

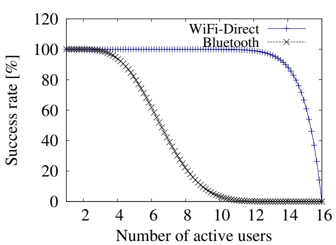

Figure 8: Number of concurrent mobile users that are supported by each communication interface

ers. Thus, the results provide insights about the mobility behavior of each user.

Naturally, the discovery process of the mobile devices in proximity induces a high computational load in a device. This is important if we consider that a user gets used to a specific routine to charge his/her phone [6]. Thus, when a power-hungry application is installed in the user's device, it may cause disturbance as the routine of the user is affected. Figure 7 compares the energy draining of a device when the discovery process functionality is active and when it is not. From the results, we can observe that the detection of devices consumes $\approx 12hours$ of battery life.

Figure 9 and 10 show the average number of devices discovered by each participant during each hour of the day via WiFi-Direct and Bluetooth, respectively. From the results, we can observe that WiFi-Direct provides the largest number of addressable devices. On average, a mobile device is always co-located in proximity to at least other mobile at anytime during the day.

We also analyze the discharge levels of the device based on user's routine. Figure 11 shows the results. We can observe that 80% of the users do not allow their mobile devices to drop below 60% of energy. This is important since users may lease his/her mobile resources to other users more on high levels of energy than on low energy levels.

Discussion

Based on the results of our experimental test-bed, we discuss the advantages and limitations of our study.

D2D discovery

There are several tools and protocols which facilitate the formation of a D2D on-site infrastructure at the application level, e.g., Bonjour[1], JXTA[2], Gnutella[3], etc. However, they are not able to dynamically monitor the frequent changes in infrastructure, given the user's mobility. As demonstrated in the evaluation section, the sensing of infrastructure in proximity is a task that harshly consumes the energetic resources of mobile devices. Thus, prediction mechanisms must infer the devices in proximity to form a D2D infrastructure. Our software sensor **Detector** is a first step towards the detection of D2D infrastructure with high energy efficiency.

D2D stability

It is a challenging task to measure the stability of a mobile device with respect to others, mainly due to privacy issues or limited energy. In other words, the device may not be visible at all times, e.g., WiFi or Bluetooth are off. In fact, recently, mobile platforms, e.g., Android and iOS, are making the process of discovery more secure for the user such that, when a device is discovered, is provided a random MAC address to identify itself to others. This introduces an extra level of complexity to estimate stability beyond our control.

While our study provides insights about the number of smartphones which are on average in proximity of the mobile user, it does not measure their temporal duration. Stability is an important factor to form infrastructure that is reliable. In other words, reliable infrastructure should meet processing deadlines during its temporal junction. Stability can be modeled based on the frequency and duration in which a device is detected. Since different devices provide different levels of stability based on the context of the user, it is possible to identify communities that can be trustful to

[1] https://www.apple.com/support/bonjour/
[2] https://jxta.kenai.com/
[3] http://rfc-gnutella.sourceforge.net/

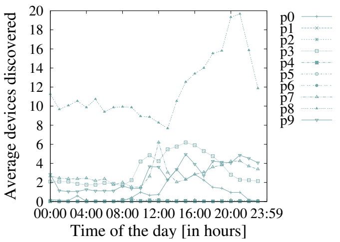

Figure 9: Average number of devices discovered via WiFi-Direct.

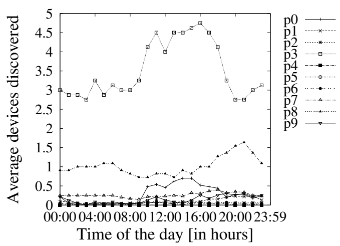

Figure 10: Average number of devices discovered via Bluetooth.

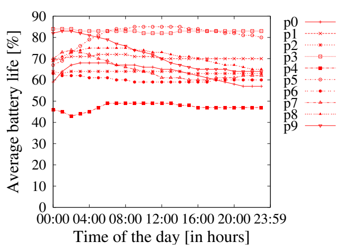

Figure 11: Average battery life of each participant during the experiment.

automate the balance of computational load among devices (D2D fog computing).

Furthermore, the identification of a stable community is important to determine the reliability of the devices, such that the most stable devices can perform analytic roles at the edge (D2D edge computing). The analysis of communities will be address in our future work.

D2D coordination

Since the communication interfaces have a limit regarding the number of concurrent requests that a device can perform, the distribution of a computational task among multiple devices is also restricted. However, it is possible to envision the design of a hierarchical dissemination approach to propagate a task. For instance, assuming that there is a D2D infrastructure of 10 devices, but one device from the group can just address successfully 6 of them to balance the computational burn of one mobile application. Then, it is possible to use a hierarchical approach, such that a parent device can address 5 devices for processing and 1 for sub-coordination. The aim of the sub-coordination is to address the 3 devices left from a second level hierarchy to delegate processing from the parent using a sub level. The design and implementation of the approach are part of our future directions.

D2D crowdsensing and crowdsourcing

Besides the sharing of processing resources among the mobile devices, the creation of a D2D infrastructure can exploit other resources of the mobile, e.g., sensors. Cross-device sensing [4] is a potential approach for augmenting the scope of a sensor by creating a mesh of sensors (big sensor), which can collect more rich and trustful information about a sensing event.

Similarly, in the context of crowdsourcing, a D2D infrastructure is utilized to disseminate the realization of task to other participants that share similar social environments. The key insight of the approach is introducing a social incentive mechanism, which can foster the participation of a high number of users by relying on social links.

Conclusions

In this paper, we design *Detector*, a sensor to detect infrastructure in proximity. We used our sensor to quantify the amount of infrastructure that is available in the wild. Our study provides insights about how much D2D communication can contribute towards the adoption of Edge and Fog Computing. Lastly, we openly share our sensor and dataset in GitHub[4].

Acknowledgements

The authors thank the anonymous reviewers for their insightful comments. This work is partially funded by the Academy of Finland (Grants 276786-AWARE, 285062-iCYCLE, 286386-CPDSS, 285459-iSCIENCE), and the European Commission (Grants PCIG11-GA-2012-322138 and 645706-GRAGE).

REFERENCES

1. Valerio Arnaboldi, Marco Conti, and Franca Delmastro. 2014. CAMEO: A novel context-aware middleware for opportunistic mobile social networks. *Pervasive and Mobile Computing* 11 (2014), 148–167.
2. Chiara Boldrini, Marco Conti, Franca Delmastro, and Andrea Passarella. 2010. Context-and social-aware middleware for opportunistic networks. *Journal of Network and Computer Applications* 33, 5 (2010), 525–541.

[4] https://github.com/mobile-cloud-computing/HybridComputationalOffloading

3. Flavio Bonomi, Rodolfo Milito, Jiang Zhu, and Sateesh Addepalli. 2012. Fog computing and its role in the internet of things. In *Proceedings of the first edition of the MCC workshop on Mobile cloud computing*. ACM, 13–16.

4. Pei-Yu Peggy Chi and Yang Li. 2015. Weave: Scripting cross-device wearable interaction. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. ACM, 3923–3932.

5. Klaus Doppler, Mika Rinne, Carl Wijting, Cássio B Ribeiro, and Klaus Hugl. 2009. Device-to-device communication as an underlay to LTE-advanced networks. *Communications Magazine, IEEE* 47, 12 (2009), 42–49.

6. Denzil Ferreira, Anind K. Dey, and Vassilis Kostakos. 2011. Understanding human-smartphone concerns: a study of battery life. In *Pervasive (Pervasive'11)*. Springer-Verlag, Berlin, Heidelberg, 19–33.

7. Huber Flores, Pan Hui, Sasu Tarkoma, Yong Li, Satish Srirama, and Rajkumar Buyya. 2015. Mobile code offloading: from concept to practice and beyond. *Communications Magazine, IEEE* 53, 3 (2015), 80–88.

8. Huber Flores and Satish Srirama. 2013a. Mobile code offloading: should it be a local decision or global inference?. In *Proceeding of the 11th annual international conference on Mobile systems, applications, and services*. ACM, 539–540.

9. Huber Flores and Satish Narayana Srirama. 2013b. Adaptive Code Offloading for Mobile Cloud Applications: Exploiting Fuzzy Sets and Evidence-based Learning. In *Proceeding of the 4th ACM workshop on Mobile cloud computing and services*. 9–16.

10. Kyunghan Lee, Joohyun Lee, Yung Yi, Injong Rhee, and Song Chong. 2010. Mobile data offloading: how much can WiFi deliver?. In *Proceedings of the 6th International COnference*. ACM, 26.

11. Yong Li, Ting Wu, Pan Hui, Depeng Jin, and Sheng Chen. 2014. Social-aware D2D communications: qualitative insights and quantitative analysis. *Communications Magazine, IEEE* 52, 6 (2014), 150–158.

12. Jakob Mass, Satish Narayana Srirama, Huber Flores, and Chii Chang. 2014. Proximal and social-aware device-to-device communication via audio detection on cloud. In *Proceedings of the 13th International Conference on Mobile and Ubiquitous Multimedia*. ACM, 143–150.

13. Rajesh Sharma and Anwitaman Datta. 2012. Supernova: Super-peers based architecture for decentralized online social networks. In *Communication Systems and Networks (COMSNETS), 2012 Fourth International Conference on*. IEEE, 1–10.

14. Yanru et al. Zhang. 2015. Social network aware device-to-device communication in wireless networks. *Wireless Communications, IEEE Transactions on* 14, 1 (2015), 177–190.

15. Yulei Zhao, Yong Li, Yang Cao, Tao Jiang, and Ning Ge. 2015. Social-aware resource allocation for device-to-device communications underlaying cellular networks. *IEEE Transactions on Wireless Communications* 14, 12 (2015), 6621–6634.