Face-to-Face Ad-hoc Networking on Android

Marcel Großmann, Lukas A. Schwöbel, Udo R. Krieger
Faculty of Information Systems and Applied Computer Science
Otto-Friedrich University
Bamberg, Germany
Email: {marcel.grossmann, udo.krieger}@uni-bamberg.de
lukas-antii.schwoebel@stud.uni-bamberg.de

***Abstract*—Mobile data consumption establishes as a massively growing market, because the proliferation of powerful smartphones and tablets on mobile networks is a major traffic generator nowadays. In order to guarantee an acceptable quality of experience to the end user, current studies try to improve data transmissions on air interfaces. However, the bottleneck in cellular mobile networks is the traversal of the infrastructure. A paradigm to circumvent this fact and to increase the system's capacity is given by face-to-face communication in combination with proximity detection. Those networks promise scalability and performance improvements in utilizing scarce resources. Without using new hardware, a face-to-face paradigm is implemented and tested with a chat application for the Android operating system. It is based on a methodology to discover devices that are in near distance to each other. Finally, a measurement trial reveals the battery drainage of smart devices in self established ad-hoc networks.**

I. Introduction

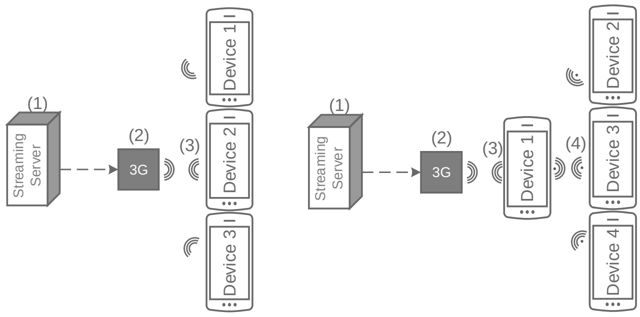
Among a few other scientific studies on mobile peer-to-peer (P2P) video streaming, Eittenberger *et al.* [1] present a prototype for a mobile P2P video streaming application running on the Android operating system (OS). The bottleneck for the video streaming of the framework and similar P2P networks is their current restriction to rely on the existing mobile network infrastructure. The connections to a P2P network, which mobile devices can establish, are mostly twofold: On the one hand, a mobile handheld can connect via an IEEE 802.11 wireless network, if the smart device provides the corresponding hardware. The WiFi infrastructure is used for short distances, dedicated to an area with static access points (AP), and offers a limited mobility to the smart device. On the other hand, cellular air interfaces are providing another entry point to the Internet that can be used to access networks. Additionally, a cellular network over relatively narrow air interfaces suffers from hand-overs between the base stations (BS), which are handled by radio network controllers (RNC), and areas that are not covered by a BS. Hua *et al.* [3] propose a multicast approach with ad-hoc transmissions. In their study the video is encoded into different layers, one base layer and multiple enhancement layers. Thus, devices with a higher distance from the BS receive less enhancement layers. The proposed ad-hoc multicast extension allows the reception of layers with a higher quality for devices that are farer away, if there are a few more peers between the BS and the receiver. Since most smart devices provide the ability to open an IEEE 802.11 AP, a proximity based approach for P2P networks is possible, such that *face-to-face* (F2F) connections can be used for data transmissions. However, this paradigm should be analyzed further to consider the battery drainage of mobile devices. In this context real-time video streaming is a good example, when a lot of users request a live video transmission at the same place. Upcoming approaches to use F2F communication on the physical layer are, for example, FlashLinQ [4] by Qualcomm or the LTE advanced approach presented by Fodor *et al.* [2] that describe methods to increase the spectrum and energy efficiency of traditional cellular networks.

We provide a first implementation of a chat application prototype that uses a F2F overlay for the exchange of messages. It is based upon the communication principle to connect nearby devices directly and thus, data transmissions are possible without further infrastructure. The prototypical implementation of a F2F chat lays the foundation to evaluate such a transmission paradigm.

II. Data Dissemination in Next Generation Networks

A. Face-to-Face Transmission

Figure 1 reveals two different data dissemination scenarios. While Figure 1a depicts the *traditional* data transmission with a BS (2) and non intelligent devices, Figure 1b shows a transmission paradigm, where one device (3) acts as an AP for a crowd of devices. Through the AP all connected devices communicate F2F without



(a) Over a mobile network (b) With F2F paradigm

Fig. 1. Data dissemination strategies

the need of a mobile network infrastructure.

Figure 2 shows a simplified diagram of the developed Android application and the communication paradigm between the activities. The *Application* stores information, e.g., the WiFi IP address, the location of the tracker-server, an identifier for the device in the F2F overlay and some device specific data. The F2F communication paradigm itself consists of two phases that are derived from the Android *Activity* class. In the first phase, the application starts the *BootstrapActivity*, which is responsible for the initialization of the application. It gathers information about the device and estimates the device's geolocation as described in Subsection II-B. The data is frequently sent to a tracker-server, which evaluates them and determines, if there are devices nearby to construct a WiFi F2F overlay. During this phase, either a WiFi AP is opened or the device connects to an already existing WiFi AP that is in near distance to the current device. Therefore, the tracker-server requests the device with the highest battery charge to open an AP and informs all nearby devices, how to establish the connection to the WiFi network.

Once a successful WiFi connection is established, this activity is stopped and remains in the background, while an instance of *F2FActivity*, the second phase, is started. Within the second activity, a F2F overlay is established that distributes data among the participating devices. To test the data exchange prototypically, the app starts a F2F chat. Furthermore, the synchronization with the tracker-server continues, such that the geolocation of the device is continuously updated. In case of losing the local WiFi connection, the *BootstrapActivity* resumes in order to setup a new connection.

B. Proximity Detection

In order to obtain the device's geolocation and to discover other nearby devices, several metrics need to be obtained. First, in the mobile context, the solution should be efficient and save battery charge. To achieve this and to guarantee a good user experience, locating the device should be fast. The longer a global positioning system (GPS) signal is obtained to get a better accuracy, the more battery power is consumed. If the accuracy is already good enough to identify nearby devices, there is no need to consume more time for a better accuracy and waste battery power. Device location is performed in multiple steps with different accuracies. Figure 3 shows three different types of information that are used by the tracker-server to evaluate the device's location and nearby devices. The cell tower (1) serves the devices with a mobile network connection like 3G. In Android it is possible to read the location area code (LAC) and the cell ID (CID). The geo-position of the cell tower is determined by using an API[1] from Google. The big dashed circle represents the accuracy of this method. As only the location of the cell tower is obtained but not the device's location, the obtained position is only as good as the size of the cell. Unfortunately, there seems to be no way to obtain the size of a cell and an accuracy of 1,000 meters is assumed. Of course, this is not precise but has some advantages. If the device is inside a building there is no way to get a GPS position, but the CID can still be obtained as long as there exists a mobile network connection. Also, the position of the cell tower is determined instantly and without noteworthy additional battery consumption. Thus, we consider it as an appropriate backup geolocation.

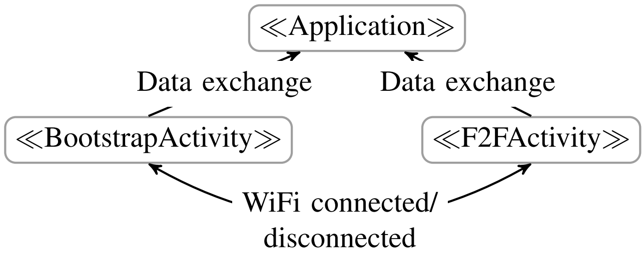


Fig. 2. Activities

As there is only a limited amount of time when the GPS position will be obtained, this position will not be accurate as well. The dotted circles (3) show the corresponding accuracy of the received GPS positions for the devices. The current approach at the tracker-server is to use this accuracy values in order to find nearby devices. Devices that are within the dotted radius of the dotted circle are assumed to be nearby. Finally, the critical point is the reachable area of the device's WiFi (2). If the circles of two devices are overlapping, they

[1] http://www.google.com/glm/mmap?mcc=xxx&mnc=xxx&towerid=xxx

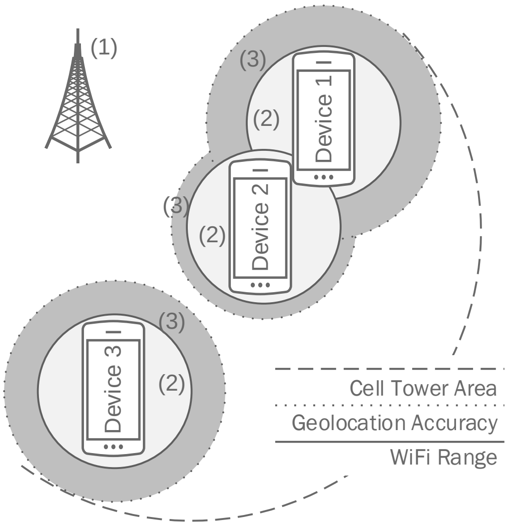


Fig. 3. Geolocation Estimation

are considered to be within each others WiFi coverage. As explained before, the tracker-server is not able to get the exact position of a device because the position is not accurate enough. The worst that happens is that one device opens up a WiFi AP without another device really being able to connect. As the protocol thereafter waits for the first device to connect, it remains in an idle state. The remaining devices would continue searching for an open AP. It only consumes and wastes more battery power at the device with the open AP. Furthermore, it seems to be safe to assume that several devices trying to establish connections at the same time in the same cell are really near to each other.

III. WiFi vs. Battery Drainage

We determined the battery drainage of two devices, the Huawei U8160 with Android 2.2 as AP and the Huawei Mediapad with Android 4.0 as client counter part. The measurement setup contained different settings with alternating bitrates and distances between the devices. Therefore, the traffic is sent from the AP to the client side in three different rates for a total time of ten minutes. In the measurements three different distances, 5, 10 and 15 meter were observed with traffic rates of 250, 500 and 1,000 $\frac{\text{KBit}}{\text{s}}$. Figure 4 depicts the total battery drainage, on the left side for the device that opens the AP and on the right side for the connected client. The total battery consumption on the client side increases for 250 and 500 $\frac{\text{KBit}}{\text{s}}$ for higher distances, but decreases for transmissions with 1,000 $\frac{\text{KBit}}{\text{s}}$. In contrast, the AP generally shows a decrease of the consumption by an increasing distance.

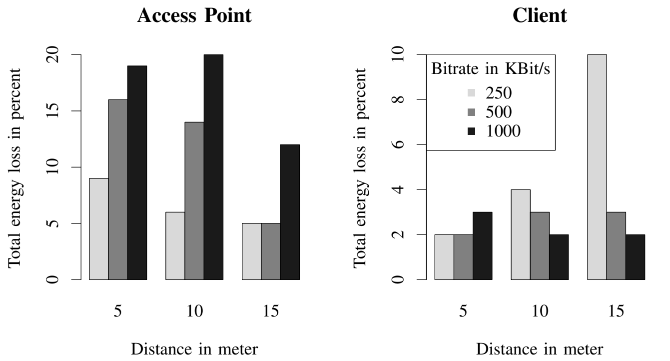


Fig. 4. Energy Loss

IV. Conclusion & Future Research

The developed prototype lays the foundation to evaluate a new F2F transmission paradigm, especially concerning the battery drainage of mobile devices. Moreover, it contains a strategy to find nearby located devices in an appropriate manner. Of course, there is plenty of room to implement new ideas, e.g., a beaconing process to find nearby devices without the need of a central controller instance. In addition, the measurement of the battery power consumption revealed a varying behavior of devices that opened an AP compared to the client counter part. Future research concerning power wastage should consider the influence factors for this observation. With regard to the transmission of media content, this could be a feasible approach to offload mobile network infrastructures, especially by enhancing mobile P2P systems with a F2F approach.

Acknowledgment

The authors would like to thank Dr. Ingo Dahm of the Deutsche Telekom AG for the provided access to the T-Mobile 3G network.

References

[1] P. Eittenberger, M. Herbst, and U. Krieger, "RapidStream: P2P Streaming on Android," in *19th International Packet Video Workshop (PV 2012), Munich, Germany, 10-11 May 2012*, 2012.

[2] G. Fodor, E. Dahlman, G. Mildh, S. Parkvall, N. Reider, G. Mikls, and Z. Turnyi, "Design aspects of network assisted device-to-device communications," *Communications Magazine, IEEE*, vol. 50, no. 3, p. 170177, 2012. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6163598

[3] S. Hua, Y. Guo, Y. Liu, H. Liu, and S. S. Panwar, "Scalable Video Multicast in Hybrid 3G/Ad-hoc Networks," in *Global Telecommunications Conference, 2009. GLOBECOM 2009. IEEE*, 2009, pp. 1–6.

[4] X. Wu, S. Tavildar, S. Shakkottai, T. Richardson, J. Li, R. Laroia, and A. Jovicic, "FlashlinQ: a synchronous distributed scheduler for peer-to-peer ad hoc networks," in *Communication, Control, and Computing (Allerton), 2010 48th Annual Allerton Conference on*, 2010, pp. 514–521.