Automatic and On-demand Mobile Network Operator (MNO) Selection Mechanism Demonstration

Christos Tsiaras, Samuel Liniger, Burkhard Stiller
University of Zurich, Department of Informatics (IFI), Communication Systems Group (CSG)
Binzmühlestrasse 14, CH–8050 Zürich, Switzerland
tsiaras@ifi.uzh.ch, samuel.liniger@uzh.ch, stiller@ifi.uzh.ch

Abstract—**The use of smartphones has been increasing in the last few years. Many open standards and accessible Application Programming Interfaces (API) make it easier for developers to achieve their ideas and many communities, such as xda developers, or stackoverflow provide good questions and answers concerning mobile application development. Questions on how to search for available MNOs and how to switch a Mobile Network Operator (MNO) programmatically on Android devices already arose in 2010. Until the work concluded here, an answer of those questions have not been published. The main reason is that there are no methods provided in the public Android API that allows for performing such tasks. In this work here two mechanisms allowing for (a) an automatic and on-demand MNO selection and (b) an MNO look-up mechanism have been developed for the Android platform. The efficiency of these mechanisms has been evaluated with respect to power and time consumption.**

Index Terms — **Android, mobile network operators, look-up mechanism, selection mechanism, energy efficiency**

I. Background

The development and the evaluation of an automatic and on-demand Mobile Network Operator (MNO) selection mechanism for the Android platform which is presented in [4] and demonstrated in this work, is primarily motivated due to the following three reasons. (a) The number of the available MNOs in a certain location, (b) the potential benefits of such mechanism for the three main mobile communication stakeholders (MNOs, end-users, regulation authorities) and (c) the existing amount, as well as the future estimation of the devices in the mobile phones market that can support such a mechanism.

II. MNO Selection and MNO Look-up Mechanism

A. Architecture

Besides the public Android Application Programming Interface (API) that is accessible with the Software Development Kit (SDK), there is also an API which is located in the package `com.android.internal` [2] that is not accessible via the SDK. While developing Android applications the `android.jar` library is referenced. In this library all classes, enumerations, fields and methods that are marked with the annotation `@hide`, from the `internal` package are removed. When the application is launched on a device the library `framework.jar` which is equivalent to the `android.jar` is loaded. However, the `framework.jar` library provide access with Java reflection to all the internal API components from the `internal` package. The `internal` package contains a class `GSMPhone` which has a public method `selectNetworkManually`. Within this work this method has been used for the automatic and on-demand MNO selection.

The class `GSMPhone` also contains the method `getAvailableNetworks` which has been used for the purpose of an MNO look-up. To get a list with the available MNOs a handler has been implemented, which is invoked once the network scan has completed.

III. Time and Energy Evaluation

Average look-up time and energy consumption determine two major dimensions of interest for an operational system on a mobile device.

A. Look-up Time

The average look-up time has been evaluated by searching available MNOs (a) in a stable position (100 times) and (b) when moving on a train from Zurich to Lucerne (300 times). Figure 1 illustrates the average time needed for a successful MNO look-up, where the error bars indicate the standard deviation. The average MNO look-up time in case of (a) was 35.71 s and 32.45 s in case of (b). The shorter average MNO look-up time when moving could be related to less available MNOs while moving. However, the standard deviation in this case is larger probably due to mobility.

The average MNO switching time has been evaluated by MNO hopping. 3600 MNO hops took place in case of (a) and 600 MNO hops in case of (b). All MNO switching times are shown in Figure 2.

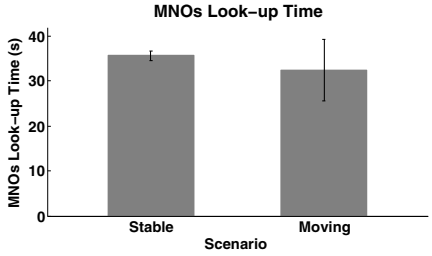

Fig. 1: MNOs Look-up Time

B. Energy Consumption

The total power consumption for the MNO look-up in a stable position was 0.123 W, while the MNO switching energy-cost was 0.540 W. MNO look-up power consumption while moving resulted 0.231 W and MNO

978-1-4799-0913-1/14/$31.00 ©2014 IEEE

switching 0.654 W. Comparing this energy consumption of the MNO look-up and the MNO selection mechanisms with a voice call on 2G and 3G networks (0.333 W and 0.705 W respectively) showed that the mechanisms implemented consume energy in the same order of magnitude as voice services.

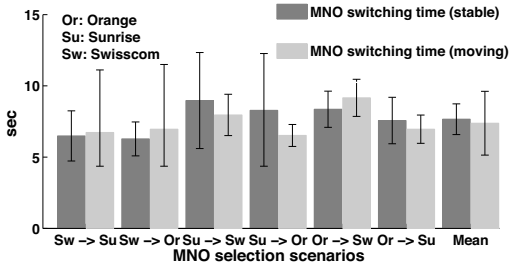

Fig. 2: Switching Time Between MNOs

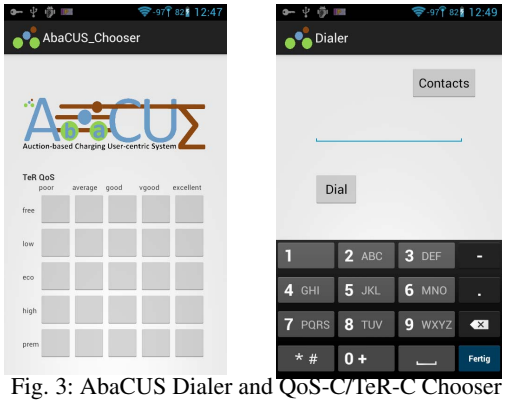

Fig. 3: AbaCUS Dialer and QoS-C/TeR-C Chooser

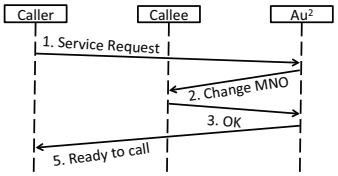

Fig. 4: AbaCUS Protocol (Service Request)

IV. Demonstration Description.

The Auction-based Charging User-centric System (AbaCUS) [3] charging mechanism propose a flexible call termination ecosystem, where MNOs participate in an auction defining which MNO will terminate a call. During this process MNOs bid on termination rates per location and per Quality-of-Service (QoS) parameters, such as the network access time and the sound quality during a call. If someone wants to perform a phone call the QoS class (QoS-C) and its corresponding Termination Rate Class (TeR-C) has to be selected first, followed by dialing the callee's MSISDN (cf. Figure 3). The high-level process of the phone call according to the AbaCUS protocol is summarized in Figure 4.

To address requirements of AbaCUS, which uses the automatic and on-demand MNO selection mechanism, an Android application has been implemented. The demonstration scenario consists of two smartphones running a custom Read Only Memory (ROM), namely CyanogenMod and an Auction Authority (Au^2) mock server, which simulates an auction, where MNOs compete to terminate a call. To simulate a flexible call termination environment, smartphones need a Subscriber Identity Module (SIM), which the local MNOs will accept. Thus, two prepaid SIM cards issued by foreign MNOs need to be used. The SIM card selected needs to be chosen with the criterion that the registration, while in roaming with every MNO in the demonstration area, is possible.

Figure 5 illustrates the demonstration set-up. MNO_1 will be the winner of the AbaCUS auction. In the beginning the Caller and Callee will be registered to different MNOs. After the successful win of the first MNO in the AbaCUS auction, the callee's device will unregister from MNO_2 and register with MNO_1. Then the call will be terminated by the new MNO.

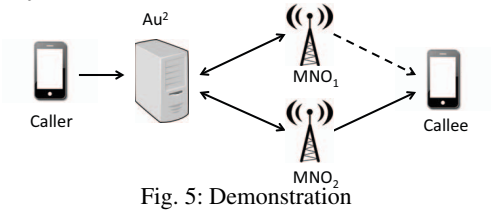

Fig. 5: Demonstration

V. Summary, Conclusions, and Future Work.

The MNO look-up and the automatic on-demand MNO selection mechanisms demand less energy than 3G call services do and the time consumption in both cases is reasonable. Therefore, such mechanisms are applicable. Therefore, the existing knowledge on how to use the internal Android API has been combined to gain access to more methods in connection with the GSM modem, which the open Android API does not provide. Thus, an automatic and on-demand MNO selection mechanism has been achieved. However, since the internal API is not listed publicly and also may change in the future, the implementation of this work is not future-safe: if new policies of MNO selection may be adopted in the future, an MNO look-up and an automatic and on-demand MNO selection mechanism has to be published in the open API by smart phone vendors. Nevertheless, the source code of the mechanism and detailed instructions are public [1].

Acknowledgements

This work was supported partially by the SmartenIT and the FLAMINGO projects, funded by the EU FP7 Program under Contract No. FP7-2012-ICT-317846 and No. FP7-2012-ICT-318488, respectively.

References

[1] AbaCUS webpage, URL: http://www.abacusproject.eu/, Visited in Aug. 2013.

[2] Android internal API, URL: https://devmaze.wordpress.com/2011/01/18/using-com-android-internal-part-1-introduction/, Visited in Jul. 2013.

[3] C. Tsiaras, B. Stiller, "Challenging the Monopoly of Mobile Termination Charges with an Auction-based Charging and User-centric System (AbaCUS)", Networked Systems (NetSys 2013), Stuttgart, Germany, March 2013, pp 110-117.

[4] C. Tsiaras, S. Liniger, B. Stiller, "An Automatic and On-demand MNO Selection Mechanism", Network Operations and Management Symposium (NOMS 2014), Krakow, Poland, May 2014.