Demo: PhoneLets – Offloading the Phone off your Phone for Energy, Cost and Network Load Optimization

Andrius Aucinas
Computer Laboratory
University of Cambridge
andrius.aucinas@cl.cam.ac.uk

Jon Crowcroft
Computer Laboratory
University of Cambridge
jon.crowcroft@cl.cam.ac.uk

ABSTRACT

This demo presents how phone functionality can be offloaded from a smartphone over wireless link to a *PhoneLet* by sharing one SIM card across multiple devices. This can lead to significant cost and network load reductions by decreasing the number of simultaneously connected mobile clients. Furthermore, it can save energy for the mobile user when connected to a powered *PhoneLet* by offloading phone functionality. It absorbs the energy cost of online presence and inefficient mobile applications' communication patterns, instead providing connectivity for the user over a WiFi link.

Categories and Subject Descriptors

C.2.3 [**Computer-communication networks**]: Network OperationsNetwork Management

Keywords

Mobile; Energy; Cellular; Networks; SIM; Subscriber Management

1. WHAT PHONELETS ARE

A modern smartphone has more in common with a computer than a phone, and connectivity to the Internet and surrounding devices (the *Internet of Things*) is tipping the balance even further. Importance of cellular networks, however, remains as they provide almost ubiquitous Internet connectivity around the world. Cellular connectivity has its own problems. This demo shows how *PhoneLets* allow one SIM (Subscriber Identity Module) to be shared between nearby devices. This addresses the following problems with cellular connectivity:

Firstly, connectivity is relatively expensive for users [4], typically requiring a separate subscription for each device connected to the cellular network, e.g. smartphone and tablet as well as broadband connectivity at home and at work. Using *PhoneLets* to share a single SIM across devices solves this issue.

MobiCom'14, September 7-11, 2014, Maui, Hawaii, USA.
ACM 978-1-4503-2783-1/14/09.
http://dx.doi.org/10.1145/2639108.2641744.

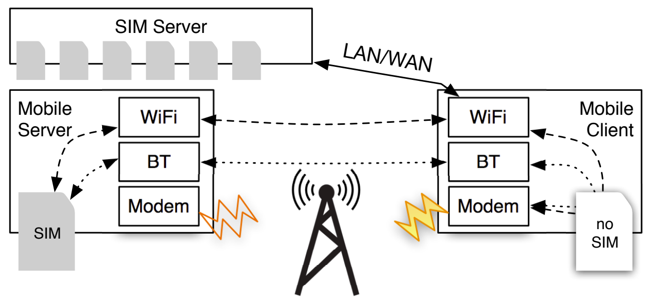

Figure 1: Architecture of PhoneLets and interaction with different types of servers

Secondly, the increasing number of connected, periodically communicating clients increases infrastructure costs and also increases cellular network load, in extreme cases causing *signaling storms* [9] that can bring down an entire network. *PhoneLets* reduce the number of simultaneously connected clients by sharing the same cellular link across multiple clients.

Finally, energy consumption of always connected clients is high compared to short-range radio technologies. Sharing a cellular link also helps reduce it for mobile devices.

PhoneLets enable phone offloading by exploiting remote SIM functionality of existing hardware. A cellular module exchange commands with a SIM/UICC module over a wireless communication link rather than direct, physical connection. Uses of such technology have included automotive as well as electronic currency [8]. *PhoneLets* infrastructure establishes and manages secure communication channels between participating entities and allows a device to connect to a cellular network without a physical SIM card.

2. PHONELETS DESIGN

Figure 1 shows the overall concept and system architecture. There are three types of entities: 1) mobile server, a mobile device which can share its SIM with other clients, 2) SIM server which may host and provide more sets of credentials over local or wide-area networks and 3) a mobile client which can use either of the methods.

The SIM card is a special I/O device in a mobile platform: due to its security requirements it is not exposed to the platform's operating system. Instead, it is managed by the modem's firmware which routes all communication between the network and the subscriber module. Standard remote IO approaches are similar in their nature (*e.g.,* RIO [6] enables

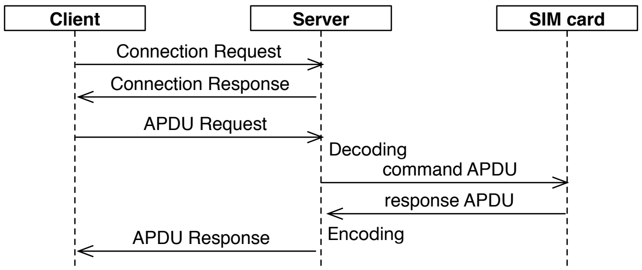

Figure 2: Communication sequence between client, server and remote SIM card

transparently accessing I/O devices of a remote mobile system) but are not applicable in this setting as they require OS support.

Interaction with the subscriber module requires lower level protocols understood by the modem firmware. Our design extends the standard RSAP (Remote SIM Access Protocol) [1] by using a SIM server which is not capable of mobile connectivity itself and by using arbitrary communication links to access a remote SIM card (*no SIM* in the mobile client). Our work for this demo focused primarily on the client and SIM server design. We use off-the-shelf hardware as the mobile server for backwards-compatibility.

The mobile client can obtain network credentials from any server – a user's mobile phone, a co-located or a remote SIM server – as long as it can reach it. The key observation is that there are few operations that require access to the SIM card due to its interface speed (it would be infeasible to use it on the data path) and instead it is used to generate encryption keys. The generating keys, however, can never be accessed directly. Part of the network's security however comes from the network authenticating the subscriber for any major operation (e.g. placing or receiving calls, establishing PDP context). The cellular module therefore must always be able to access the remote card using standard Application Protocol Data Unit (APDU) messages (Fig. 2).

Importantly, we do not clone the SIM card and by design the system disables connectivity for one device before enabling it for another one to prevent multiple devices sharing the same credentials simultaneously. Only one device is connected to the cellular network and may share (*tether*) the connectivity to other devices over another channel. One cellular module is also only capable of being registered with one network, enforcing a strict one-to-one mapping.

2.1 Prototype

Our demo mobile client (Fig. 3), the *PhoneLet*, consists of a Telit UC864 cellular module connected to a raspberryPI as a controller, which also uses WiFi and BlueTooth (BT) dongles for local wireless connectivity. The controller runs a modified version of the Ofono open telephony stack and manages communication between the cellular module and multiple backends.

The two example servers are an off-the-shelf Samsung Galaxy S4 and our custom server that only interfaces with SIM cards through a smartcard reader, also running on a raspberryPI. In our prototype, the server is connected to a WiFi AP to be reachable by the mobile client. We replace the BT communication channel with a more secure option - TCP link with certificate-based authentication of both client and server, and strong encryption for all data exchange. Once the client establishes a connection to either backend, it registers with the network and takes on the role of a phone.

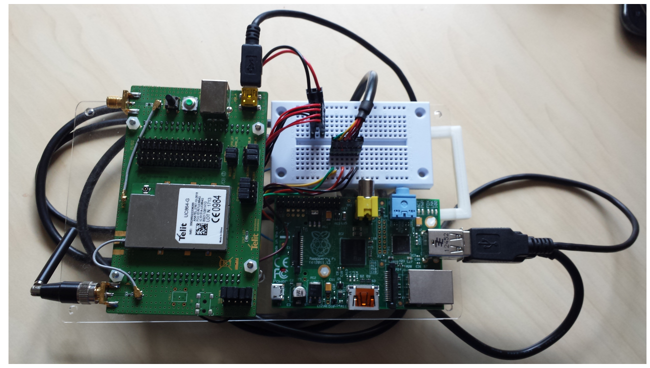

Figure 3: Client prototype

Use case	Energy consumption (mA)
Idle - airplane mode	5
Ifle - BlueTooth	6
Idle - WiFi	8
Idle - 3G	10
Apps background - WiFi	27
BlueTooth SIM access	28
Apps background - 3G	40
WiFi tethering (idle)	80

Table 1: Energy consumption of an idle device with different radios enabled. (Apps background = Google apps and Facebook)

3. ENERGY SAVING OPPORTUNITIES

Reduced mobile client energy consumption is an important motivation for *PhoneLets*. We analyze energy consumption of the client in a few example scenarios using our early prototype. The *PhoneLet* itself is not yet energy-optimized, i.e. the controlling RaspberryPI is not meant for mobile uses, so we focused on energy comparison of the mobile SIM server - a phone carried by the user from which we offload phone functionality.

We measured energy consumption of a Samsung Galaxy S4[1] in a number of scenarios. Table. 1[2] presents energy consumption of the different device radios, depending on their use, ranging from an idle device with only a particular radio enabled, to different methods of sharing connectivity and simple application usage patterns.

The absolute values will vary with different devices and network configuration or conditions and application use, but the overall results remain. In particular, energy consumption of the different idle radios is similar, but cellular networks are significantly less efficient for current applications' network usage patterns [5]. Therefore, replacing the cellular link with a WiFi one to another, powered device can bring significant savings - 32.5% in our example case.

[1]Using Monsoon Power Monitor `http://www.msoon.com/LabEquipment/PowerMonitor/`

[2]Current drain (mA) is chosen over power (mW) as power depends on voltage that decays with the battery discharge

The default BT implementation of RSAP is not efficient enough - maintaining access alone consumes almost three times the amount of energy of an idle device connected to a 3G network. Instead, we have implemented SIM sharing over WiFi using our custom client and verified that the BT version is unnecessarily wasteful - once the PhoneLet has registered to the network, there is no communication with the SIM server for as long as the device is idle. For example, a phone call is successfully received after 30 minutes of no activity. The current Android platform does not allow RSAP to run over WiFi, but based on our measurements we expect energy consumption of an optimized implementation to be close to that of an idle WiFi channel - 3.5x less than BT.

Finally, we measured energy consumption of WiFi tethering. With just one idle client it is almost 3x higher than the BT case. RSAP is effectively an alternative to WiFi tethering when the mobile client provides connectivity to other devices. Therefore we expect energy consumption of an optimized WiFi mobile client to be close to 10x less than WiFi tethering.

These are early results, but they point to a potential use-case where a powered *PhoneLet* is registered to the network on behalf of a WiFi-connected client at a considerable energy benefit. The benefit would be further improved when network access is shared across multiple such devices and SIM access load is balanced across them as the savings for each individual device would add up. This use-case is outside the scope of the current demo.

4. DEMO: WIRELESS SIM SHARING

To demonstrate the working prototype of the system, we show our client *PhoneLet* (Fig. 3), communicating with an off-the shelf mobile device or our SIM server to share their credentials for network registration in the following steps:

1. Make a phone call to the mobile server (phone) and show that it is ringing
2. Start sharing its SIM card with the client and show that the client is registered to the network with the same ID by making a phone call to it
3. Start using SIM card in the SIM server over WiFi and show that the client is registered to the network with a different ID by making a call to the other number.

In the three steps we show our prototype's ability to offload phone functionality from a phone - share subscriber credentials between multiple clients and select which one will be connected to the network.

5. DISCUSSION

In addition to connectivity, *PhoneLets* offers management of an individual's ID - the phone number. There have been multiple offerings with such function: multiple SIM cards associated with a number or VoIP-based solutions (*e.g.,* Google Voice), however they do not solve the problem of an increasing number of cellular clients and billing depends on service provider.

One cellular network in UK has offered a service to make phonecalls from tablets and laptops over WiFi to its users using a special application[3], blurring the line between mobile and the Internet. If WiFi Internet connectivity is available, the approach could help reduce energy consumption by disabling cellular radio.

It is well known that WiFi tethering consumes a lot of energy. Cool-Tether [7] proposed to reverse the roles of a mobile gateway and client and turn the client into WiFi AP. Our proposal also reduces energy consumption of the mobile gateway by completely changing the roles of the entities. Instead, the client (*PhoneLet*) uses the cellular network directly, avoiding additional wireless communication with the gateway.

Reprogrammable SIM cards [2] such as embedded SIM [3] have been proposed for provisioning in M2M communication. They maintain a physical secure element to store credentials, but provide mechanisms to manage them over the air. The architecture allows operators to retain full control over the credentials, including when new ones can be installed or old ones removed through centralized management and subsequently reduced flexibility on the client side.

Finally, questions that remain to be answered in future work include not only optimizations of the management and control protocols, but also the development of usage policies in multi-user environments and guaranteed isolation properties as well as adaptation of energy-aware routing protocols for distributed nodes.

6. SUMMARY

Increasingly, smartphones are replicating computer use, with phone now a very secondary function to email, social media and entertainment. However mobile network connections remain expensive both financially and in terms of the energy they use. *PhoneLets* is a system that allows a single mobile subscription to be shared between multiple devices, allowing much of the mobile functionality of a cellular device to be offloaded. This allows the financial and energy costs of a cellular connection to be shared between multiple devices.

More information and demonstration video is available via `phonelets.smart-e.org`.

7. REFERENCES

[1] SIM Access Profile. Technical Report SAP_SPEC, Bluetooth SIG, 2008. V11r00.

[2] Reprogrammable SIMs: Technology, Evolution and Implications. Technical report, CSMG, OFcom, 2012.

[3] Embedded SIM Remote Provisioning Architecture. Technical Report 2FAST.13, GSM Association, 2013.

[4] EU27 mobile data cost competitiveness report. Technical report, Rewheel Consulting, 2013.

[5] A. Aucinas, N. Vallina-Rodriguez, Y. Grunenberger, V. Erramilli, K. Papagiannaki, J. Crowcroft, and D. Wetherall. Staying online while mobile: the hidden costs. In *CoNEXT '13*, 2013.

[6] A. A. Sani, K. Boos, M. H. Yun, and L. Zhong. Rio: a system solution for sharing i/o between mobile systems. In *MobiSys '14*, 2014.

[7] A. Sharma, V. Navda, R. Ramjee, V. N. Padmanabhan, and E. M. Belding. Cool-Tether: energy efficient on-the-fly wifi hot-spots using mobile phones. In *CoNEXT '09*, 2009.

[8] H. C. Subramanian. Sim access profile: Electronic currency using sim access profile. 2003.

[9] C. Yang. Weather the signaling storm. Technical report, Huawei, 2011.

[3] O2 TU Go `http://www.o2.co.uk/apps/tu-go`