Mobile Application Profiling for Connected Smartphones

Almudena Diaz, Pedro Merino, and F. Javier Rivas

Abstract

The Smartphone application market is expected to continue to grow at even higher rates **and keeping high levels of quality will be a key factor for this success**. The community of **mobile** developers is attracting members from other environments with mobile version initiatives for languages like Python or Ruby. But developing mobile applications requires facing mobile specific constraints and issues. With current development environments and profiling tools, it is highly difficult to deal with mobile communication issues: the main problems appear when using actual devices in real operations instead of emulators. We introduce a tool that aims to fill that gap by providing developers with the means to analyze mobile data communications. **The key novelty of the profiling tool introduced in this paper is the correlation of traffic information and radio access technologies measurements with location data, which helps third party developers to test and evaluate their mobile applications on the field**.

New and innovative applications for mobile devices have emerged in recent years due to the appearance of open operating systems and virtual machines optimized for mobile devices [1]. On one hand, programmability of mobile devices paves the way for a new generation of applications which takes advantage of the intrinsic mobility, the proximity to the user and its dependence on hand held devices. On the other hand, software development for mobile devices also involves new trends and problematic issues [2] that need to be tackled to assure the sustainability of the smartphone application market.

New smartphones are equipped with powerful multi-task processors. Some of them also provide real-time guarantees, but the processing capacity of traditional desktop computers, as well as the performance, differ a lot from those provided by mobile devices, so developers face new constraints in this restricted and unknown development environment. These constraints are not only related to hardware issues such as processors or memory but also, as we will explain in this paper, to the lack of specific tools for debugging the different aspects that affect the performance of mobile applications.

Bluetooth, WLAN, GPRS/EDGE, UMTS/HSDPA or WiMAX are examples of technologies available on mobile devices. The use of these wireless technologies introduces new variables in the development process which are unknown by developers accustomed to work on fixed networks. Each one of these technologies has different behaviors and is suitable for different contexts. **In fact the use of different radio access technologies at the same time is a new challenge for mobile developers**. There are lots of applications which allow analyzing the performance of mobile applications in terms of resource consumption, such as memory or processing time, but the performance of communication in mobile applications have not been taken on from the developers point of view. That is, mobile developers do not have available usable tools to answer questions regarding

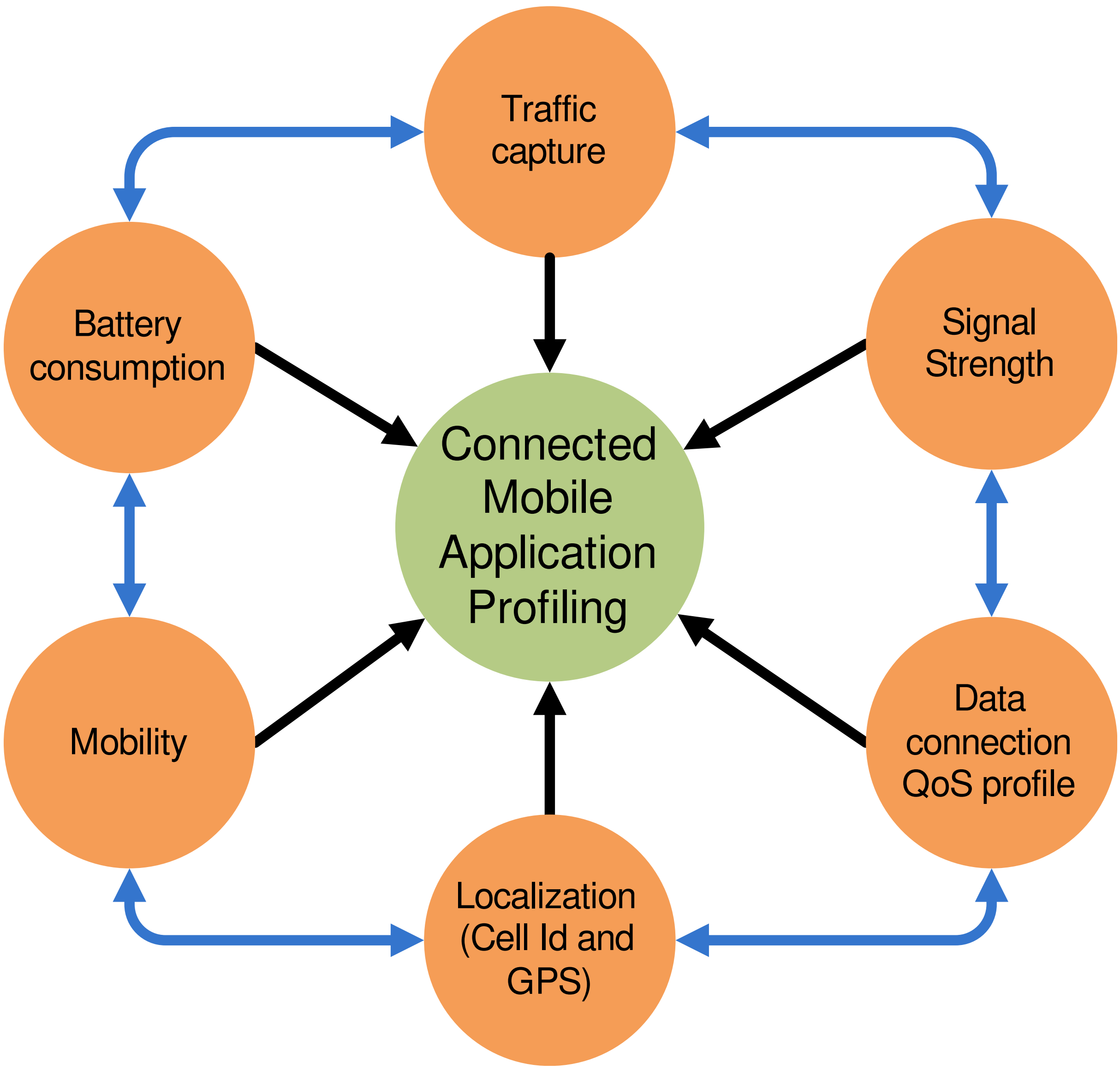


Figure 1: Mobile Application Profiling

communication performance of their applications: How can I debug the behavior of the data connections used by the application? What is the performance of the protocol I have implemented? Can I improve the performance of the connections established by my applications?, What is the quality of the connection used by my application?, What is the real available bandwidth that I can use? These kinds of question proliferate in dynamic scenarios where propagation conditions are more unstable, and vertical handover between different radio access technologies takes place.

To provide the answers to these and other questions, and to help developers find the cause of errors related to accessing network resources, we have developed SymPA, a software tool which adds packet sniffing features to mobile devices, localization data, as well as other relevant functionalities depicted in figure 1, which allow profiling **mobile** applications with advanced communication capacities.

SymPA has been successfully tested on different projects carried out at the University of Malaga, ranging from the development of mobile applications to the characterization of mobile services over cellular networks [3] [4].

In addition, other universities and companies have contacted with us to use SymPA in the development of their product. VectorMAX, a company specialised in the development of on-demand video solutions, is using our tool for debugging their applications. PacketVideo and several international mobile operators have tested the tool and have demonstrated their interest in the next version of SymPA for Symbian 9.

Currently, we are working on the integration of SymPA and QoSMeT [5], a passive measurement tool for measuring one-way end-to-end network QoS (Quality of Service) developed for the Technical Research Centre of Finland (VTT). The integration of both tools will enable one-way mobile device measurements in real time.

In this paper we will show why the functionalities provided by SymPA will play an important role in the improvement and the debugging of the increasing number of mobile applications.

1 State of the art

SymPA has been developed over Symbian OS, an operating system specifically designed for mobile phones with an ample range of supported programming languages (Symbian C++, native C, native C++, Java, Python, Flash Lite, Ruby), which allows testing applications programmed in different languages.

The study "Converged smart mobile device market: smart phones and wireless handhelds" carried out by the analyst firm Canalys shows that Symbian leaded the smartphone operating system market on Q4 2007 with 65% share, ahead of Microsoft on 12%, RIM on 11%, Apple on 7% and Linux at 5%. Moreover in the growing Symbian application market there are nearly 10,000 commercially available third-party applications, entailing a 25% increase over the previous year. Although penetration level differs depending on the region, Symbian leads the sales in regions such as Europe, Japan or China, whereas RIM leads the market in North America

As this study indicates, Symbian is nowadays the most extended operating system for smartphones. In addition, it provides access to low level libraries which are necessary for the implementation of an application of the characteristics of SymPA. In the future, Linux based platforms appears as a very promising platform for porting the tool because they will inherit a wide variety of applications and libraries related with traffic analysis, although it is expected that some mobile specific features will not be covered by previously existing tools. Due to this, we are carrying out the tracking of the evolution of Linux-based platforms such as OpenMoko, Maemo and Android and cross-platforms solutions such as Qt extended, for porting SymPA and enabling the testing of the applications developed over them.

There are other Symbian tools for wireless network performance analysis, such as QualiPoc (www.swissqual.com), Nemo Handy (www.anite.com) and m-Formation (www.mformation.cor but, in contrast to SymPA, they are totally oriented to network operators. They provide low level information and center on specific service tests, but they do not allow third party developers to debug the behavior of the protocols they implement or to solve the connectivity problems they have, or correlate these issues with location data, which is extremely valuable.

Other solutions focus on providing on target tests and debugging capabilities. Application debugging in real devices is indispensable because applications usually work successfully on the emulator, but fail on real targets. It is also one of the most time consuming tasks, so the on target test is one of the stages of the development process which needs to be simplified and facilitated. The Remote Device Access (RDA) service provided by Forum Nokia offers remote access to a wide range of smartphones, so developers can test their applications on a pool of real devices. RDA also provides the possibility of testing on prototypes, which enables developers to design new applications taking advantage of the functionalities incorporated in these new devices, and new devices and new applications could be delivered to the market at the same time. Other similar initiatives from Forum Nokia are the Virtual Developer Lab, powered by DeviceAnywhere and the Device Loaner Program. These initiatives reveal the importance of using real devices during the development of mobile applications for smartphones.

Furthermore, during the development process testing phase, it is essential to deploy the application on real devices in order to analyze the performance of the application on the real hardware for which was designed. Nokia Energy Profiler (NEP) enables developers to monitor their applications' energy usage in real-time; in addition the processor, memory and network signal levels are analyzed. The performance investigator delivered with Carbide.c++, a Eclipse-based development tools supporting Symbian OS development, enables on-target data tracing of function calls, power usage, memory usage and key events. Both tools focus on analyzing the hardware performance of the applications, CPU usage, memory availability, or battery consumption. **But evaluation of mobile applications not only requires using the actual mobile device but also field trials [6]. That is a key issue in the development process of any mobile application and which is covered by SymPA.**

Finally, we refer to some specific services which deal with security and stability issues. Stability is key for smartphones, since they are devices which must be operative for long periods of time. For example, they must always be able to switch away from any application in execution and make an Emergency Call. Symbian Signed (www.symbiansigned.com) is intended to provide a unified testing and certification process with the objective of ensuring that devices continue to operate normally after the applications are installed or removed, that applications are safe for end users and also that applications do not harm the operation of the network. This initiative, and others such as Java Verified (www.javaverified.com) or True Brew Certification (http://brew.qualcomm.com), reveal that application integrity concerns not only end users but also developers, mobile network operators and phone manufacturers.

After the introduction of these tools and services, it is clear that the different stakeholders of the mobile sector are aware that the correct performance of mobile applications is vital for the technological evolution of devices, applications and services. But it is also clear that the performance of the communications, an open issue in the application development process [7], is beyond the scope of current development support tools.

2 New profiling needs in "always connected" applications

Development of mobile software for smartphones has large timescales [8] since programmers need to manage new APIs and operating systems and also new implementation methodologies. In the previous section we introduced a set of tools for supporting different aspects of the development process. However, some relevant aspects, like communication

Figure 2: SymPA menus

performance, are not sufficiently covered. Improving resource consumption and checking stability issues is key to obtain high-performance applications, but networking problems are also one of the most important issues developers deal with during implementation and testing phases on real devices.

Guaranteeing the networking performance of mobile applications is key for the future of smartphones. Moreover, one of the reasons this new generation of mobile phones has been designated as "smart" phones is because of their capacity to communicate with a large number of smart networks, which range from near field communication (NFC) to global communication using cellular technologies.

Networking issues on wireless networks and on mobile phones differ a lot from networking issues on fixed networks of traditional desktop computers. They require much more efforts on the part of developers because of the lack of traditional tools to check connectivity, to analyze the traffic data, to study the configuration of the protocol stacks implemented on mobile devices and to debug performance errors related to mobile communications. For example, the implementation of servers on mobile devices is a tedious task due to limitations imposed by operators in their networks. In this situation developers should be capable of discerning between a problem related to closed ports and a problem related to the implementation of the server itself.

This issue may be trivial, but more complex issues related to the changeable nature of mobile communications also need to be analyzed to discern between application performance issues or network errors. Moreover, developers should be able to cope with issues such as link outage, but to implement this kind of functionality they need tools with which to analyze traffic and correlate it with location, speed and wireless network issues. Battery consumption of the different radio technologies is also a concern of mobile developers, who always strive to improve the battery consumption of their applications.

The configurations of the protocol stacks available on mobile devices are different from the configurations of a traditional personal computers, so the use of emulators is discarded. The use of the mobile phone as a modem is not recommended because applications tested this way use the computer protocol stack to establish data connection. That means that traditional protocol analyzers can not be used to study networking performance. So if we want to analyze connectivity issues, the only solution is to execute the applications

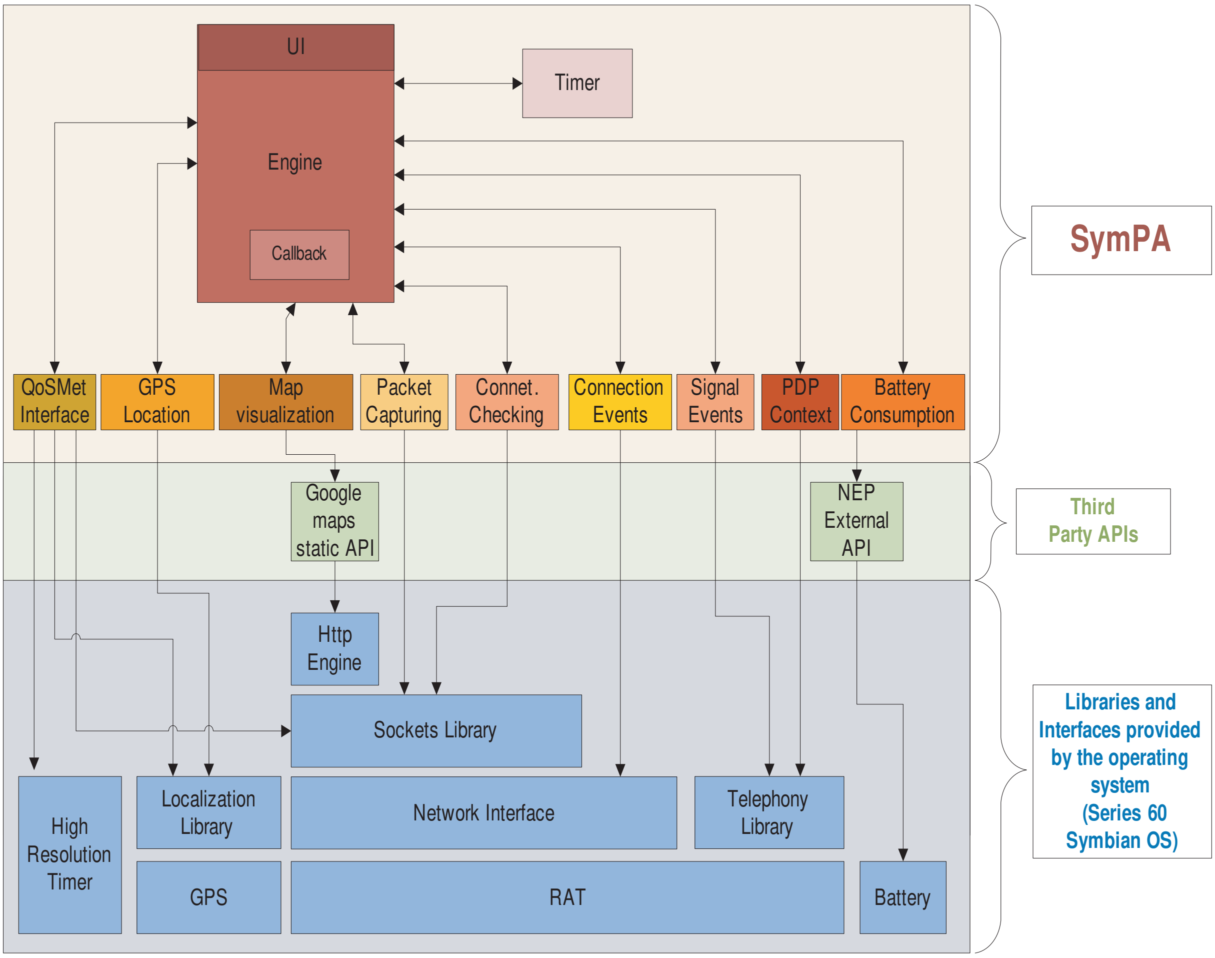


Figure 3: SymPA architecture inside the smartphone

on real devices. Therefore we need tools which also run on real devices. **The SymPA profiling tool helps developers to deal with these issues. Figure 2 shows some screenshots of the application running on a Nokia 6110 Navigator.**

Coding errors can be detected using tools for on-device debugging, but the under performance of communications can not be detected with this kind of tool. Development and testing times would be reduced if we had had adequate tools for debugging the behavior of the protocols implemented.

Heterogeneous environments is also a special scenario where the use of the tool introduced in this paper could be relevant. The possibility to work over heterogeneous networks is one of the most interested possibilities offered by mobile devices. It is a very promising advantage to be exploit by mobile developers due to it allows the development of always connected applications. But, from other point of view, heterogeneous environments are critical scenarios where dealing with connectivity could become a challenge for mobile developers.

In this scenario SymPA enables the evaluation of the performance of vertical handover on mobile devices which should make possible a seamless roaming between different networks. SymPA could be used for analyzing the impact of these techniques on IP communications from the point of view of the application. In particular, SymPA has been used for characterizing handover

between GPRS and UMTS networks. In figure 6(a) we can see the burst of lost packets during a handover on a streaming session in a vehicular test. Also signal strength, jitter, bandwidth and battery consumption are represented for each technology.

3 Enabling the analysis of mobile applications

The functionalities provided by SymPA have been extracted from the knowledge obtained in of all the projects we have been involved over the last years.

Figure 3 shows its architecture and the different modules implemented to provide these functionalities, which will be explained in more detail in the following sections. In the figure we can also see, the libraries and the application program interfaces used during the implementation of the tool. These libraries and interfaces have to be provided by the operating system in order to implement an application of this type.

SymPA functionalities are centered on analyzing networking issues. In general radio propagation conditions become unstable in wireless scenarios, where factors like signal quality and received signal strength, which are affected by doppler and multipath fading effects, trigger cell reselections and handovers between different access technologies. Analyzing that information is key to identify the source of connectivity issues such us call drops, packet losses or bandwidth reduction. SymPA helps to monitor all this parameters.

GPS tracking and map visualization functionalities are also available for enabling the localization of the measurements in static and dynamic scenarios. These two functionalities are key for the profiling of **mobile** applications, not only to correlate performance measurements with localization data but specially to obtain the mobility patterns of users.

As we will see in the following sections, the profiling information provided by SymPA can be easily integrated in applications such as Wireshark or Google Earth.

3.1 Analyzing data connections on mobile phones

Three main features are provided to enable data connection analysis: traffic capture, received signal power strength tracking and radio access technology identification, also providing the QoS profile to cellular networks during connections.

Data traffic capture is the main functionality provided by SymPA. This powerful feature is fundamental not only for traffic analysis, but also for debugging communication protocol implementations. A wide range of measurements can be applied to the connections established to obtain parameters such as bandwidth, delay and jitter. Other useful utilities include connection filtering and incoming establishment availability check. This is specially adapted to identify whether a mobile operator has closed required ports, as it is a usual practice aimed at blocking specific types of traffic.

Raw captured data traffic is stored in the memory of the mobile terminal, also offering an option to export data to libpcap format. Since libpcap is a widely used standard format, it is suitable to be displayed in widespread analysis tools such as Wireshark.

3.2 Checking connectivity

SymPA enables developers to check the connectivity of the mobile phone.

Mobile to Internet connectivity may be verified by establishing data connections and then obtaining the IP address associated to the connection. Obtaining the IP address is useful, as it varies each time a connection is established and may be utilized for further data processing or to accept connections in servers running in smartphones.

But it is in the Mobile to Mobile use case where checking the connectivity becomes more useful, because it is not possible to deploy traditional tools at the server side as in a typical Internet scenario. For this purpose the application allows the mobile devices to transfer files acting both as client or as server to accept incoming connections. This active traffic generation utility, in conjunction with the traffic capture feature, enables mobile to mobile connection performance evaluation.

Also a standard Ping functionality is available to provide an estimation of the round trip delay in connections. A proper characterization of communication delays is required to design timers and buffers adapted to the actual conditions that mobile applications will experience.

3.3 Measurement localization

Two new features have been added to accurately locate the measurements collected and to know the associated speed for each individual piece of information. The GPS tracking function activates the monitoring of the speed of the mobile terminal as well as its position. All the GPS related information is stored in a file, e.g. timestamps and coordinates. Timestamps will allow matching different information sources as captured traffic and radio condition information in a later post-processing stage.

Through the use of Google Static Maps API it is also possible to visualize directly on the mobile device a map of the area (see figure 5(a)). With this information it is easy to identify roads, buildings and other objects that could have an effect on the communication.

In addition, through the post-processing of the data files generated by SymPA (see figure 4) it is possible to generate more meaningful and detailed maps, including coverage and handovers over the path followed by a user.

In figure 5(b) we can see an example of how location information provided by the GPS tracking functionality allows a deeper understanding of communication issues related to user mobility. The red line in the graph represents the received signal power during a bike ride from the VTT Technical Research Centre of Finland in Oulu to the city centre while accessing an audio streaming service. The blue vertical lines with numbered marks on top correspond to changes from one cell to another. To generate this representation we have used the Google Earth application. During the experiment, at some point near the river in the figure we experienced degradation in the quality of the sound and a temporary interruption. After finishing the bike ride, with a first analysis of the captured information we could easily identify that the problems were not related to the server side or network congestion, but rather to a deep loss of received signal strength at the boundary between two cells. In the figure it is evident that the degradation occurs around the change to cell 450787, and that after crossing the river the signal strength returns to acceptable levels.

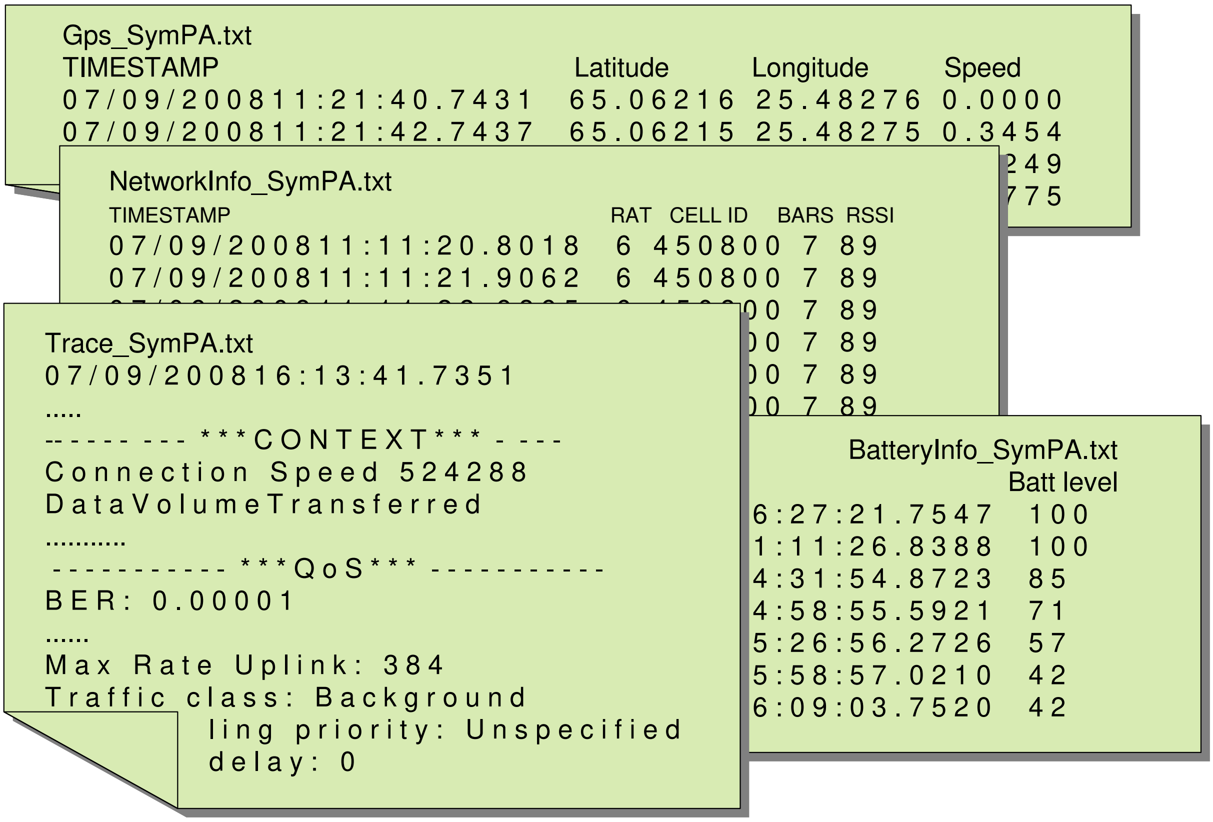


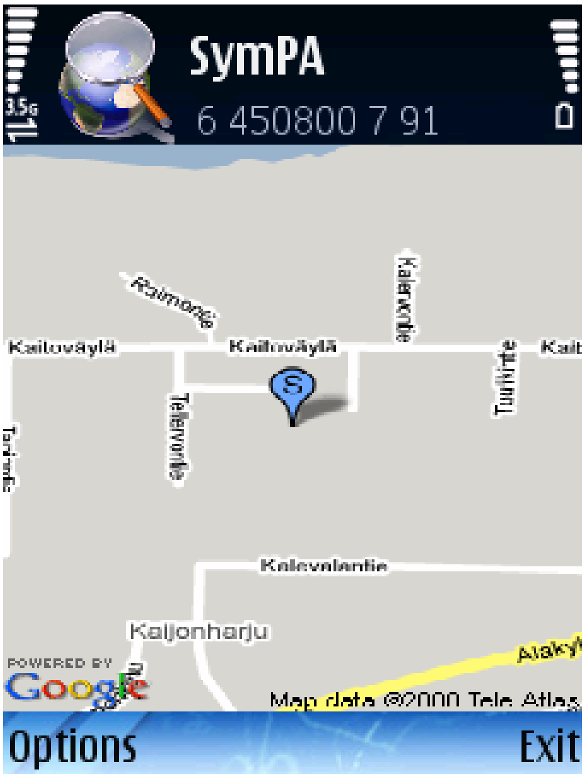
Figure 4: SymPA Profiling Files

3.4 Battery Consumption Profiling

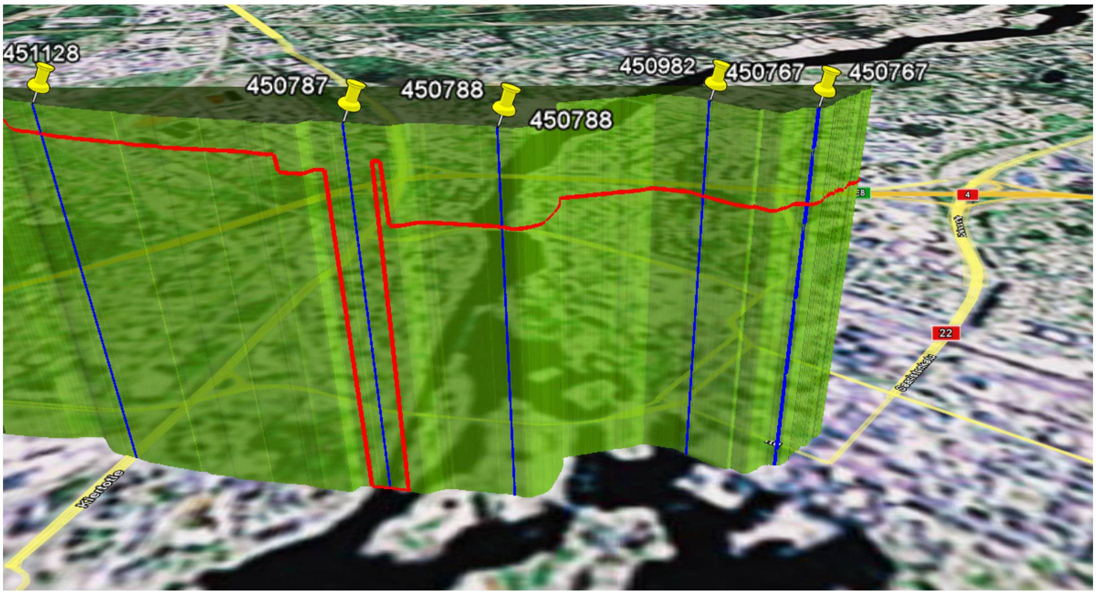
The adoption of energy saving strategies is required in mobile devices, as smartphones are battery driven devices [9]. As a common engineering practice, we need to pinpoint the sources of bigger energy consumption, to concentrate later optimization work on those critical functionalities. For that purpose we have added battery monitoring features.

But the highest benefits of this feature result from the fusion with all the other available information, as it is possible to associate the results to the actual conditions during the measurement. In traditional profiling tools, the energy related information is obtained regardless of the spatial information, which may strongly affect the results.

For example, if we compare only in a static environment the power required in a WLAN connection to the power in a UMTS connection, the results could lead to confusion, as the uplink power in UMTS may vary from insignificant values up to 2 Watts depending on the capabilities of the device, the distance to the base station, and the propagation conditions [10]. **Figure 6(a) shows the effect of GPRS/UMTS handover on energy consumption during a streaming session. As expected energy consumption is higher in UMTS and also consumption increase is more significant when received signal strength decreases. This functionalities enables third party developers to evaluate battery consumption of their applications using different radio access technologies in each particular location and context.**



(a) Geographical localization on the mobile device



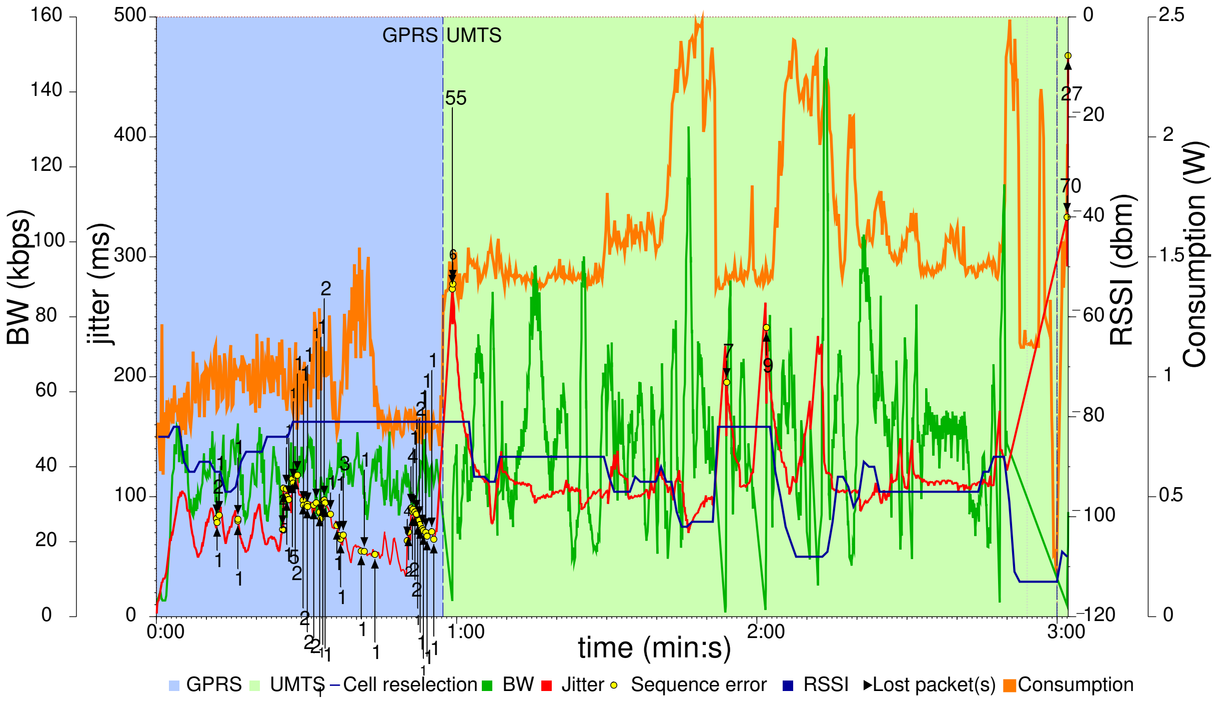
(b) Localization data post-processing: Correlation of localization, cell and signal strength

Figure 5: Measurements' localization

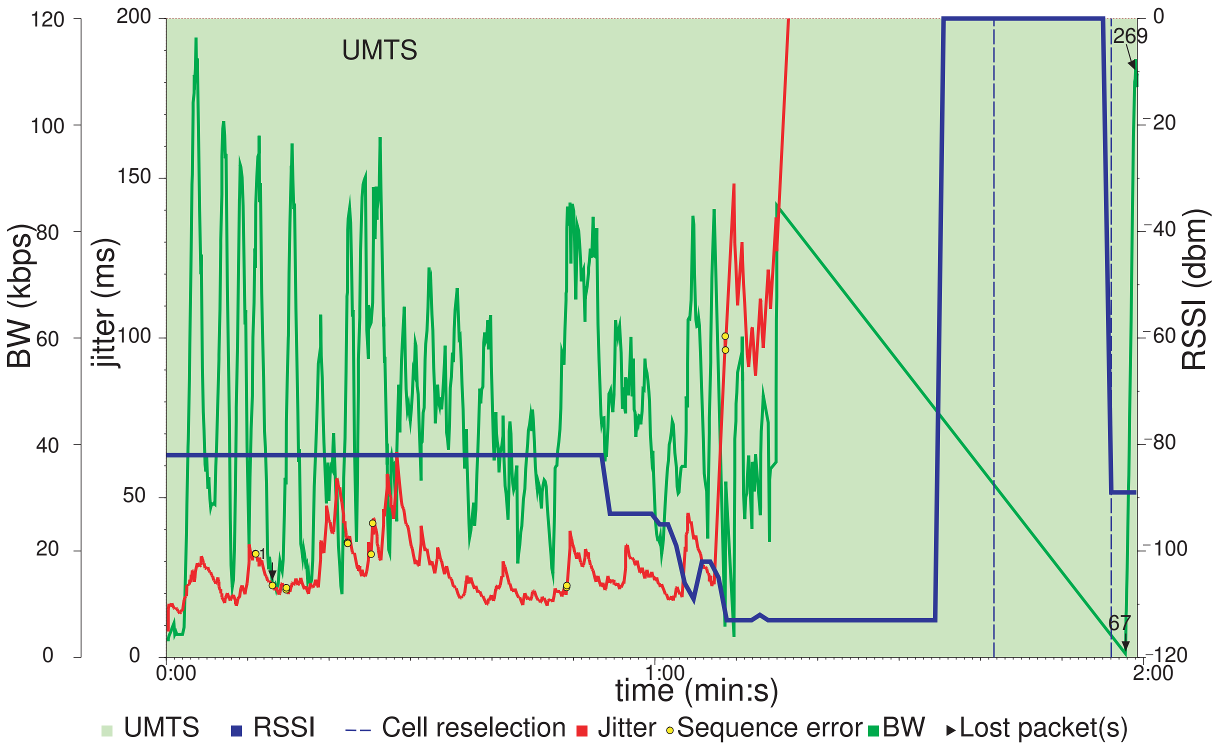
3.5 Data correlation

In this section we provide an example of how the correlation of the data obtained at different levels allows us to find the source of an error experimented at the application level.

In this case we were carrying out the profiling of a video streaming client over a public cellular network in a vehicular scenario. In the analysis of the test campaign we



(a) GPRS/UMTS handover



(b) Link outage during a streaming session

Figure 6: Analyzing data communication performance on mobile devices

detected a streaming session that finished abruptly, so we used the information collected by SymPA to detect the source of the problem. Streaming quality depends on four main factors: bandwidth, packet loss, delay and jitter. To calculate theses factors we analyzed RTP traffic captured during a streaming session with Wireshark, a well known protocol analyzer. After that, we correlated this information with signal strength and cell data. Figure 6(b) shows the results of the correlation. Signal strength experimented a strong degradation below -110 dBm, which is a typical sensitivity threshold for mobile devices. Then the signal was completely lost during 20 seconds. This link outage caused the buffer of the video streaming client to underflow and the application finished its execution.

Adding location info to this correlation process, we determined that this error took place when the vehicle was circulating in a tunnel.

This is an example of the potential provided for SymPA and how data correlation allows developers to discover the origin of errors and associate them with geographical features.

4 Conclusion

In the past few years, we have been involved in multiple projects and have obtained a first hand view of the issues and challenges that mobile application developers must face with networking analysis, debugging and profiling mechanisms. Today there are smarter devices for which we need smarter tools and solutions. We have developed SymPA with the goal of helping developers deal with these tasks.

We are currently working to provide at the application level a framework for isolating networking issues, so third party developers can take advantage of this framework and incorporate to their applications the networking facilities it provides.

The mobile market is shifting its focus from devices and communication technologies towards applications. Technology will no longer be a target but an enabler for applications that could match the needs of the huge number of mobile users.

This change of philosophy has been made possible by a new generation of smarter phones powered by open operating systems that provide support for an increasing number of virtual machines and runtime environments. In this line, we should highlight the Symbian efforts to attract to the mobile environment large developer communities through the inclusion of languages such as Python or Ruby, which are interpreted languages that allow rapid application prototyping.

With our ongoing work we aim to simplify the development process for mobile applications by providing communication awareness so that applications could exploit all the intelligence available in the new smarter phones.

SymPA is available free at www.lcc.uma.es/∼pedro/mobile/. The application has been tested on Nokia Series 60 Symbian smartphones from S60 2nd Edition initial release (Symbian 7.0s) to S60 3rd Edition, Feature Pack 1 (Symbian 9.2). Manuals and videos with information about the tool are also available at the web site.

Acknowledgments

This work has been funded by the Spanish government sponsored project TIN 2005-09405-C02-01 and by project P07-TIC-03131, funded by the Autonomous Community of Andalusia.

References

[1] F. H. P. Fitzek and F. Reichert, *Mobile Phone Programming: and its Application to Wireless Networking.* Springer, August 2007.

[2] A. Ocampo, D. Boggio, J. Munch, and G. Palladino, "Toward a reference process for developing wireless internet services," *Software Engineering, IEEE Transactions on*, vol. 29, no. 12, pp. 1122–1134, Dec. 2003.

[3] A. Diaz, P. Merino, L. Panizo, and A. M. Recio, "Experimental analysis of peer-to-peer streaming in cellular networks," in *Proc. 21st International Conference on Advanced Information Networking and Applications AINA '07*, 21–23 May 2007, pp. 784–791.

[4] A. Diaz, P. Merino, L. Panizo, and A. Recio, "Evaluating video streaming over gprs/umts networks: A practical case," *Vehicular Technology Conference, 2007. VTC2007-Spring. IEEE 65th*, pp. 624–628, April 2007.

[5] J. Prokkola, M. Hanski, M. Jurvansuu, and M. Immonen, "Measuring WCDMA and HSDPA Delay Characteristics with QoSMeT," *Communications, 2007. ICC '07. IEEE International Conference on*, pp. 492–498, June 2007.

[6] T. Halonen, J. Melero, and J. R. Garcia, *GSM, GPRS and EDGE Performance: Evolution Toward 3G/UMTS.* New York, NY, USA: Halsted Press, 2002.

[7] K. A. Brown, "Impact of wireless communication on multimedia application performance," in *Proc. SPIE Vol. 3528, p. 566-573, Multimedia Systems and Applications.*

[8] D. Wood, *Symbian for Software Leaders: Principles of Successful Smartphone Development Projects.* Symbian Press, May 2006.

[9] T. Zhang, S. Madhani, P. Gurung, and E. van den Berg, "Reducing energy consumption on mobile devices with wifi interfaces," *Global Telecommunications Conference, 2005. GLOBECOM '05. IEEE*, vol. 1, pp. 5 pp.–, Nov.-2 Dec. 2005.

[10] "User Equipment (UE) radio transmission and reception (FDD)," 3GPP, Tech. Rep. TS 25.101, May 2008.