Field measurements of mobile services with Android smartphones

Andrés Álvarez, Almudena Díaz, Pedro Merino, Fco. Javier Rivas
Departamento de Lenguajes y Ciencias de la Computación
University of Malaga
Malaga, Spain
Email: {aalvarezm, almudiaz, pedro}@lcc.uma.es, franciscojavier.rivas@gmail.com

***Abstract*—The proliferation of mobile Internet services for smartphones makes it more necessary than ever to use measurement tools in mobile devices to monitor their connectivity performance. In this paper we introduce TestelDroid, a software tool for Android devices which provides advanced monitoring functionalities for analyzing multimedia services and applications specifically designed for smartphones.**

The beneficiaries of this tool are mobile operators, service providers and mobile developers, as TestelDroid offers a unified and independent methodology to analyze performance parameters from the application level, taking into account underlying issues.

I. INTRODUCTION

Smartphones are becoming a major platform for the execution of Internet services as more powerful and less expensive devices are becoming available. The verification of the performance of such services is progressively moving from the traditional PC plus modem setup to smartphone built-in scenarios. Testing configurations based on computer based tools are not enough anymore for Internet services on mobile phones. They are giving the way to measurement tools specifically designed for smartphones, enabling performance analysis of services and applications specifically designed for mobile devices. In this transition, it becomes critical to develop powerful tools providing native measuring capabilities for mobile based data services. Thus, a wide range of monitoring functionalities are needed to allow the fusing of information from many different sources and view points, obtaining a complete profiling of mobile applications and services.

Following this approach SymPA [1] was developed and successfully applied in the study of video streaming service in cellular networks [2]. With TestelDroid we pretend to increase the monitoring functionalities with the features provided by Android and extend our research into Android services, applications and devices.

Mobile operators require scalable monitoring solutions because network complexity and mobile subscriber use patterns are increasingly challenging. Providing smartphones with advanced measurement capabilities paves the way for distributed and scalable monitoring systems as the dependency on dedicated expensive hardware is avoided. This scheme could complement and enhance network self optimization procedures extending the observation points to additional protocol layers and to the user device and applications perspective.

The first tools designed to operate in mobile devices, such as Qualipoc [3], were centered on the observation of service accessibility parameters such as availability, provided bandwidth, error rate, etc. To that end, they included active client features ranging from FTP (File Transport Protocol) to voice calls and message sending. These clients were used to test the services and verify general parameters. However, the rising complexity of mobile networks and applications requires test solutions to go a step further both in application and radio performance monitoring. By doing so, it will be possible to identify the actual sources of communication issues and to improve the user experience, as subscribers do not care whether communication problems are related to application, protocol or radio aspects, since they all result in experience degradation. Measuring solutions aiming to deal with many perspectives must therefore correlate information from all the communication layers.

Although active service probing is useful for some purposes, it is clearly unfeasible to include clients for every single application and service available, present or future. Passive monitoring represents a scalable alternative for general purpose analysis of mobile applications' performance. Different aspects may be studied jointly using the passive approach including communication performance, memory usage and battery consumption among others.

In this paper we will introduce the usability of TestelDroid in the detection of VoIP traffic behaviors associated to cellular environments. In section II we will introduce the features provided by TestelDroid for traffic capture, cellular network monitoring, GPS measurement localization and power consumption. Some noteworthy implementation details are also included. Section III presents its applicability in the analysis of the performance of multimedia services in live cellular networks providing a qualitative analysis of VoIP quality over GPRS (General Packet Radio Service), UMTS (Universal Mobile Telecommunications System) and HSPA (High Speed Packet Access) technologies. Finally, section IV provides a summary of the main contributions of this work.

II. TESTELDROID FEATURES

TestelDroid is meant to be a tool for monitoring parameters and traffic in Android based devices, in order to characterize the behavior of cellular communications. The information

978-1-4577-2071-0/12/$26.00

TABLE I
FEATURES PROVIDED BY TESTELDROID

Network	Operator RAT (Radio Access Technology) CID (Cell Identification) LAC (Location Area Code) RSSI (Radio Signal Strength Indicator) PSC (Primary Scrambling Code)
Neighboring cells	PSC RSSI RSCP (Received Signal Code Power) RAT (Radio Access Technology)
Battery	Battery level Temperature (C) Voltage (mV) Current (mA)
GPS	Longitude Latitude Altitude Speed
IP traffic	Pcap format Arrival timestamps Promiscuous mode
Connectivity test	Ping Open ports
Active traffic test	Server-Client mobile-to-mobile Transfer of auto-generated file Bit rate monitoring Average transfer speed

retrieved can be logged and exported for further analysis with other tools, such as Wireshark. Table I provides a summary of the features provided by the tool.

All the collected data can be logged using highly analyzable plain text files (except for traffic capture, stored on *pcap* format). Logging is implemented as an Android service, thus it can be running on background while performing other actions on the phone. The parameters to be logged (network, neighbor cells, battery, GPS, traffic) are configurable under the preferences menu, as shown in Figure 1. The tool has been tested on Nexus One (HTC), Nexus S (Samsung) and Galaxy S (Samsung) devices with Android versions ranging from 2.2 (Froyo) to 2.3.4 (Gingerbread). During the development of the tool we noticed that PSC (for the current cell), current consumption and neighboring cell information was missing on Samsung phones.

A. Implementation

The information provided by TestelDroid is collected at three different levels, as shown in Figure 2.

At the application framework level (blue area), Android API provides information such as current cell related information, available neighbors or GPS location. An event oriented approach is used to notify of changes such as signal strength variations, handovers, or GPS location updates. Battery data is also provided at this level except for current consumption, which will have to be retrieved from the Linux file system. Google Maps API is used as well to have a visual representation based on GPS data.

Android runtime level (green area) provides part of the connectivity test (check whether a socket can be opened at

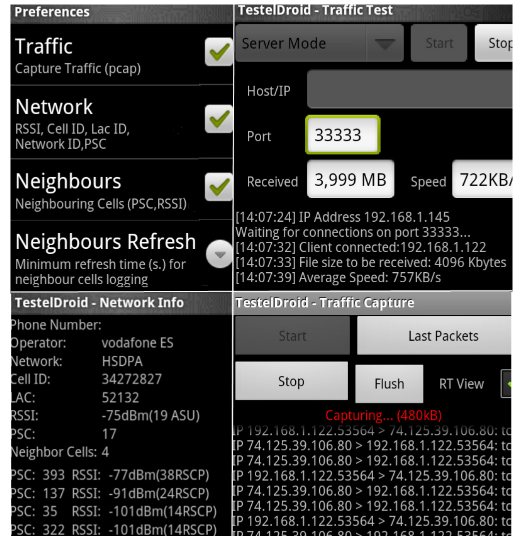

Fig. 1. TestelDroid screenshots

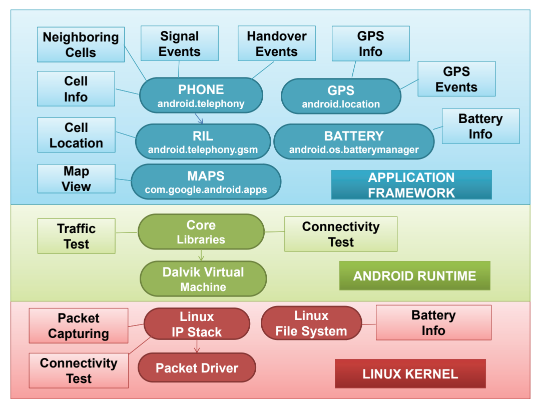

Fig. 2. TestelDroid diagram

a specified host/port) and traffic test (transfer of a file on a client/server model) features.

Finally, some tasks are accomplished directly at the Linux kernel itself (red area), such as packet capturing, part of the connectivity test (ping) and current consumption information for the battery.

III. CASE STUDY: VOIP OVER HSPA AND INTER-RAT MOBILITY

This section focuses on the evaluation of VoIP performance over HSPA at the application level, although during the experiments the RAT was eventually changed to UMTS or GPRS, depending on the coverage. There is a great deal of related work in the literature which covers this topic, but usually it is based on simulations and focused on air interface capabilities, MAC-hs functionalities such as fast packet scheduling, link adaptation and HARQ (Hybrid Automatic Repeat Request) [4], and RRC (Radio Resource Control) performance such as discontinuous reception (DRX) cycles [5]. These works exhaustively study packet scheduling and resource allocation,

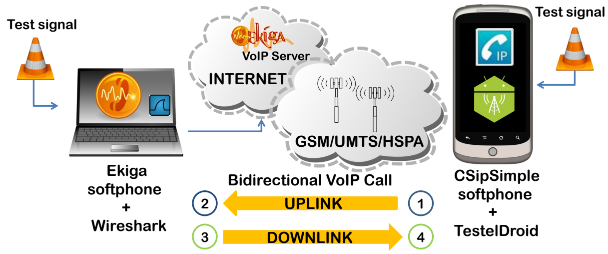

Fig. 3. VoIP experimental scenario over live cellular networks

Fig. 4. VoIP traffic transmitted by Ekiga softphone during a conversation with a mobile connected through a HSDPA connection (Downlink source side)

and obtain relevant results from the point of view of network management. However the performance of the results obtained at the application level and from the point of view of individual users has not been proved. Parameters such as IP bit rate, inter-packet delay, jitter, IP packet losses and sequence errors have not been evaluated in referenced works. Although these parameters have been the subject of study in [6] and [7], such studies are oriented to prove the performance of scheduling algorithms but by means of simulations.

Studies carried out in [8] and [9] show a close similarity with the work introduced in this paper, with the singularity that we provide an open methodology which enables studying the performance of any application taking into account radio propagation particularities. They perform test campaigns on live HSDPA (High Speed Downlink Packet Access) networks and study jitter, packet loss and end-to-end delay. Both works prove their validity to obtain relevant statistics from live testing in cellular networks, but it is worth noting, at this point, that a remarkable contribution of our paper is the possibility of evaluating the impact of radio network management procedures over mobile application and protocols performance by the simple inspection of the results provided by TestelDroid. In the following sections we will prove the soundness and the complementary functionalities offered by our tool for supporting field measurements on Android mobile phones.

A. Field trials setup

In order to verify TestelDroid in a mobile deployment, the tool was used to measure VoIP performance over live cellular networks. Both static and vehicular scenarios were covered. For the test we used a commercial VoIP service provider, Ekiga.net, which also offers a PC softphone client application. Figure 3 shows the experiment setup, where two VoIP clients were configured to establish a bidirectional communication. Data collected were post-processed and most relevant results are depicted in figures in the following sections.

TestelDroid was deployed on a Nexus One mobile phone running in parallel with CSipSimple, a VoIP client for Android. The other side of the connection was hosted by a PC running the Ekiga softphone. Both the CSipSimple and the softphone were configured to use Ekiga VoIP service. The Ekiga softphone auto answering functionality was enabled to allow automated VoIP call tests. The PC was placed in the campus network of the University of Malaga and thus connected to Internet using high speed data access. All the IP traffic exchanged during the experiment was captured using Wireshark in the computer end and TestelDroid on the mobile side. Four measurement points allow them to measure downlink and uplink traffic at both ends, as indicated in Figure 3. To provide a reference source for the experiment, a standard audio sample was injected in both devices. The sample was extracted from the artificial voices described in the recommendation ITU-T P.501, which specifies test signals which are applicable for several purposes in telephonometry. Using reference sample signals allows the analysis of the impact of impairments observed in IP-based networks, and will enable us to provide objective speech quality metrics in future experiments.

B. Static characterization of VoIP traffic

In a first step we characterized the traffic generated by both VoIP clients in a static scenario, before running the measurement campaigns in the vehicular scenario. Traffic inspection allowed us to determine the codec negotiated for both sides of the conversation, which in this case was G.711. We have observed that there are no changes in the codec used during the call once it is established. In the remaining experiments we have forced the codec to stay the same to ensure the results are suitable for comparison.

The size of the packets sent by the Ekiga client installed in the PC is 214 bytes, with a time period between audio packets of 20 ms with a small dispersion, and a fraction of the packet at 10 ms intervals. We can appreciate this in Figure 4, where inter-packet delay is depicted with orange points. This dispersion introduced a small jitter at the source (red line). Transmission rate is represented by the green line, which has an average value of 80 kbps, as we could expect because it is close to the rate obtained with 214 bytes packets at 20 ms. This is one of the standardized bit rates for G.711 codec. During the experiments the traffic bandwidth is calculated by accumulating the length of the packets received during the last second.

Figure 5 shows the parameters which define the VoIP stream transmitted by CSipSimple in HSDPA. TestelDroid facilitated the analysis of RTP (Real-time Transport Protocol) traffic

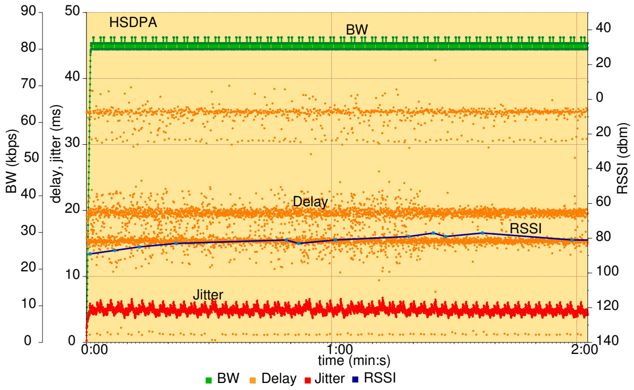

Fig. 5. VoIP traffic generated by CSipSimple softphone during a HSDPA connection (Uplink source side)

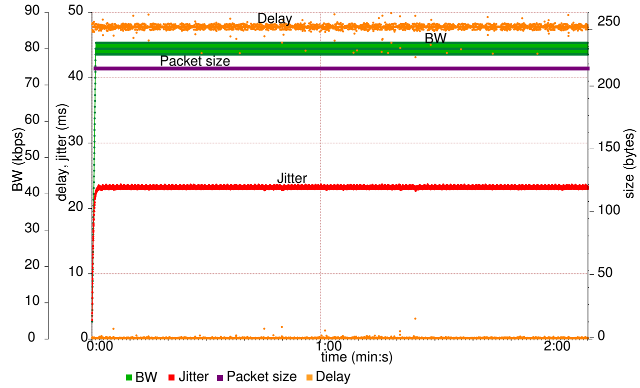

Fig. 6. VoIP traffic transmitted by Ekiga softphone during a conversation with a mobile connected through a GPRS connection (Downlink source side)

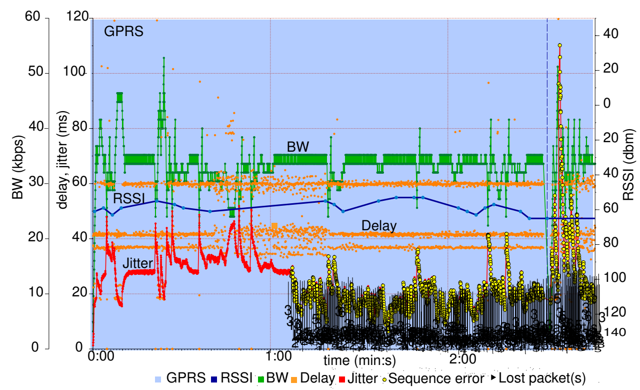

Fig. 7. VoIP traffic received by CSipSimple during a GPRS connection (Downlink receiver side)

emitted, providing access to useful information not normally available on mobile devices. In this analysis the inter-packet delay was observed to be restricted to a discrete number of values. These values differ from those observed at the PC softphone: 0, 15 and 20 and 35 ms, and present higher dispersion than previously had been observed. This dispersion introduced a jitter of 5 ms seconds at the source that must be taken into account in the analysis of the traffic on the receiver side.

The packet size and the transmission bit rate average are 214 bytes and 80 kbps respectively, which are equivalent values to those from the Ekiga client. Figure 5 also depicts RSSI (blue line) and RAT in use (orange background color indicates HSDPA). Adding information not only from the IP level, but also from the radio connection will enable us to explain some traffic patterns. As we can see in the following figures, radio parameters have a great impact on traffic parameters and without this information users would miss the actual source of the behavior observed.

In order to validate the results obtained from the information collected by TestelDroid we have compared our mean jitter results with values obtained in previous work. Averaging the results obtained during the static measurements we have obtained a mean jitter of 6 ms for HSDPA and 13 for HSUPA (High Speed Uplink Packet Access), which are in the same range than the results obtained in [8] and [9]. In the vehicular scenario we obtained a mean jitter of 23 ms but results show a high dispersion due to packet loss introduced by cell changes and fading. This mean jitter is also coherent with those provided in [8].

When the available bandwidth is reduced in GPRS, we have been able to observe that the signal quality is dramatically impacted because the codec requires higher transmission rates. Thus, in GPRS connections we have seen that the bandwidth may be reduced to 35 kbps (see Figure 7) whereas the analyzed VoIP clients do not react to this variation in the communication channel capacity, as per Figure 6. In consequence, we can observe in Figure 8 that the encoded waveform sent by the Ekiga client at the PC cannot be received at real time and it is expanded because of the increasing delays. The figure contains three waveforms: the original transmitted signal (bottom), the reconstructed waveform using timestamps to compensate the transmission delays (middle) and the actual received signal (top) which can be clearly identified as expanded because it accumulates more than 500 ms delay increments in just 5 seconds. However, because of the difference in transmission and reception rates (80 kbps and 35 kbps respectively), the network can not afford to buffer the downlink traffic indefinitely. After 80 seconds of reception in the downlink, 40 seconds of delay is accumulated which indicates that the buffering has reached its limit (more than 3 Mbits is being performed by the network) and lots of packets are dropped (marked with black arrows in Figure 7) interleaved with received packets leading to a drop rate of nearly 50% on average. If no buffering had been performed by the network, the drop rate would have increased to 56%=(80-35)/80, but the huge delay would have been avoided. It must be noted that as VoIP is a very interactive service, large delays are not acceptable. Although no additional figures are included, in the analysis of the uplink scenario associated to the downlink session in Figure 7, a similar behavior has been observed but with sequences of bursts of more than 100 lost packets.

It clearly shows that applications must negotiate quality of

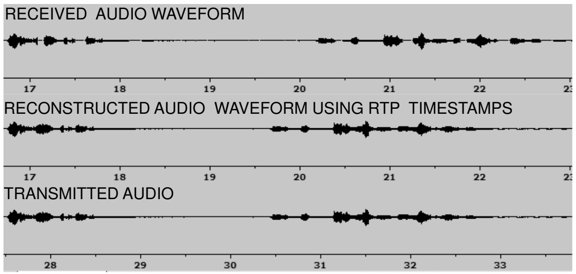

Fig. 8. Transmitted and received waveforms comparison in HSDPA

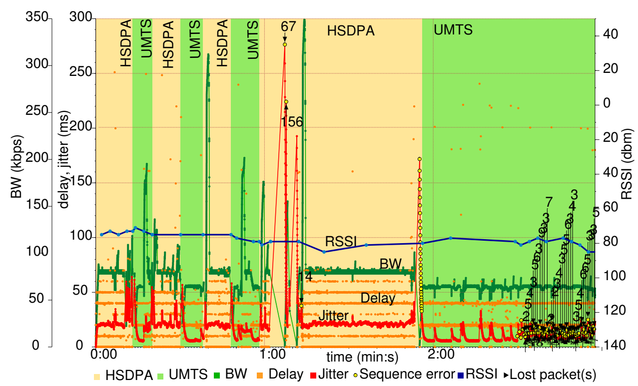

Fig. 9. Radio access technology impact over inter-packet delay (Downlink receiver side)

service parameters tailored to their needs and that the network must commercially provide connection profiles adapted to real time services.

C. Vehicular characterization of VoIP traffic

One of the observations we can make on the basis of the vehicular results is the impact of HSDPA and UMTS over inter-packet delay. The behavior observed during the tests is depicted in Figure 9. Inter-packet delay presents a clear difference in its behavior if we compare UMTS intervals (green background) and HSDPA intervals (orange backgroud).

During a vehicular test where data connection changed several times from UMTS to HSDPA we can appreciate how values acquired show a different behavior. In the case of HSDPA six values can be discerned ranging from 0 to 50 ms in steps of 10 ms. A lower inter-packet delay granularity was expected in HSDPA, as its TTI (Transmission Interval Time) duration is only 2 ms, but a similar result was also obtained in [9] during jitter calculation. When compared to the results in [9] the instantaneous jitter is not directly comparable because we calculate it as an average of the delay (following the formula in RTP RFC) which results in a low pass filter of the signal that hides the high frequency fluctuations. In our study case we can appreciate the same behavior for the inter-packet delay. That paper pointed out that it was related to the Iub flow control and the management of data buffers used between nodeB and RNC.

Regarding the packet losses and the end of session shown in Figure 9, we cannot conclude that UMTS is not suitable for VoIP because it could reach speeds of up to 384 kbps. However, by inspecting the information provided by TestelDroid we have seen that only a 64 kbps resource allocation has been dedicated to the connection by the network. In consequence a similar issue as the one introduced in subsection III.B appears when network buffer overflows. Real time interaction is also heavily impacted upon because of the high buffering delays, that reach up to 11 seconds and finally result in packet losses.

IV. Conclusion

We presented TestelDroid, a software tool which runs on Android phones. It offers a high configurability of the monitoring functionalities provided. The paper also provides multiple measurements in different scenarios, carried out to characterize the performance of VoIP in HSPA and inter-RAT with legacy technologies. A correlation of all the monitored parameters was presented to prove the potential of the tool in troubleshooting tasks. We also observed an interesting dependence between inter-packet delay behavior and radio access technology in use.

In future work, we will further develop TestelDroid monitoring features in order to extend the inspection capabilities of radio management protocols.

TestelDroid will be available through our web site: http:// www.lcc.uma.es/∼pedro/mobile and Android market.

Acknowledgment

This work has been partially funded by the Government of Andalusia under grant P07-TIC3131, the Spanish Ministry of Innovation and Science (MICINN) under grant TIN2008-05932 and FEDER from the European Commission.

References

[1] Diaz A.; Merino P.; Rivas F.J.; "Mobile Application Profiling for Connected Mobile Devices", IEEE Pervasive Computing, vol.9, no. 1, 2010
[2] Diaz A.; Merino, P.; Rivas F.J.; "QoS Analysis of Video Streaming Service in Live Cellular Networks", in Computer Communications Elsevier, Vol. 33, Issue 3, 2010
[3] Qualipoc handheld, Web site: http://www.swissqual.com/index.php/systems/qualipoc_handheld.html
[4] Wang B.; Pedersen, K.I.; Kolding, T.E.; Mogensen, P.E.; "Performance of VoIP on HSDPA", IEEE 61st Vehicular Technology Conference (VTC-Spring), 2005
[5] Aho, K.; Repo, I.; Nihtila, T.; Ristaniemi, T.; "Analysis of VoIP over HSDPA Performance with Discontinuous Reception Cycles", Sixth International Conference on Information Technology: New Generations (ITNG), 2009
[6] Weimin X.; Ghosh, A.; Schaeffer, D.; Downing, L.; "Voice over IP (VOIP) over cellular: HRPD-A and HSDPA/HSUPA," IEEE 62nd Vehicular Technology Conference (VTC-Fall), 2005
[7] Shreevastav, R.; McGoldrick, C.; Huggard, M.; "Delivering improved QoS and cell throughput in UMTS based HSDPA networks," IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks & Workshops (WoWMoM), 2009
[8] Arjona, A.; Westphal, C.; Yla-Jaaski, A.; Kristensson, M.; , "Towards High Quality VoIP in 3G Networks - An Empirical Study," 4th Advanced International Conference on Telecommunications (AICT), 2008
[9] Jurvansuu, M.; Prokkola, J.; Hanski, M.; Perala, P.; , "HSDPA Performance in Live Networks," IEEE International Conference on Communications (ICC), 2007
[10] Lunden, P.; Aijanen, J.; Aho, K.; Ristaniemi, T.; "Performance of VoIP over HSDPA in Mobility Scenarios", IEEE Vehicular Technology Conference (VTC-Spring), 2008