MANAGING TETRA CHANNEL COMMUNICATIONS IN ANDROID

Luis Corral, Alberto Sillitti and Giancarlo Succi
Center for Applied Software Engineering, Free University of Bozen-Bolzano
Piazza Domenicani 3, I-39100 Bozen-Bolzano, Italy

Keywords: Android, Mobile, Tetra.

Abstract: Mobile systems have evolved to a level in which they have to carry out their operations in environments demanding high availability and short response time. An excellent radio transmission protocol is necessary to provide the means to satisfy such requirements. TETRA standard offers a large number of features designed for safety and emergency use; however, to access a TETRA network it is compulsory the use of a standard-specific device. In this paper we explore the conditions that have to be fulfilled to extend Android's capabilities to access TETRA networks from a general-purpose mobile device, and set a path for a future development aimed to accomplish it.

1 INTRODUCTION

Mobile systems have evolved to a level where they are required to carry out their operations in environments demanding high availability, efficient performance and short response time. Along with regular radio, cellular telephone communication has been the choice of organizations that have to satisfy strict mission-critical requirements (Bissmeier, 2006).

In this context, Terrestrial Trunked Radio (TETRA) was specifically designed by the European Telecommunications Standards Institute to meet the needs of Professional Mobile Radio (PMR), primarily used in the public safety sector: police departments, emergency services, fire departments, surveillance of marine traffic and others (Bakaric, Borzic, Bratkovic, Grga, 2005).

Major manufacturer Nokia states that now TETRA is a global standard for PMR communication in the same way that the Global System for Mobile Communications (GSM) is the standard for mobile telephony (NOKIA, 2004). TETRA provides the capability of trunked, non-trunked and direct mobile-to-mobile communication. The services it offers include voice (group or private calls), circuit switched data, packet switched data and Short Data Service (SDS) - similar to Short Message Service (SMS) in GSM (Axiotis and Xenikos, 2007). TETRA also supports generic and tailored data applications for each user organization, such as command and control systems, vehicle location, database queries, reporting, image transfer and other customized applications. Thanks to these characteristics, TETRA is now the official European Standard for digital PMR and has been selected by numerous organizations in Asia, Middle-East and South America for operations considered safe and emergency critical, according to The TETRA MoU Association (TETRA MoU, 2011).

Similar to GSM, to access a TETRA network it is compulsory the use of a device designed for the use of the standard. This means that the baseband of the device should recognize the range of frequencies in which TETRA communications are transported. Currently, TETRA product lines offered by manufacturers are not as wide as other equivalent GSM equipment, and the range of prices makes disadvantageous to select a TETRA-specific device against another comparable GSM-specific unit.

In this paper we study the conditions that have to be fulfilled to access TETRA networks from a general-purpose mobile device, based on the possibilities given by Android to incorporate APIs for utilizing GSM-specific phone features. Then, we trace a path for a future development aimed to design and implement an extension to Android for supporting communication on TETRA networks.

The rest of this paper is organized as follows: section 2 presents background on current devices

Corral L., Sillitti A. and Succi G..
MANAGING TETRA CHANNEL COMMUNICATIONS IN ANDROID.
DOI: 10.5220/0003534303070312
In *Proceedings of the 13th International Conference on Enterprise Information Systems* (ICEIS-2011), pages 307-312
ISBN: 978-989-8425-55-3

available on market; section 3 introduces the motivation to conduct this research and the scenario that this work aims to solve; section 4 comments related work; section 5 describes a comprehensive analysis of the state of the question and explores the opportunities given by concurrent technologies; section 6 identify directions for future work and finally section 7 draws the conclusions.

2 TETRA MOBILE DEVICES

TETRA devices are designed and produced to work in a spectrum of radio frequencies from 300 through 1000 MHz. In Europe, TETRA frequencies are mainly allocated in the 380 to 430 MHz, keeping lower radio frequency band (380-385 MHz and 390-395 MHz) reserved for emergency and public services and the higher (410-430 MHz) for civil use (Bakaric, Borzic, Bratkovic, Grga, 2005).

The number of active TETRA projects has continuously shown a growing trend (Mikulic and Modlic, 2008). With the increasing number of networks, organizations and users, there is as well an increasing need of development of technology (e.g. infrastructure, devices, accessories) to interact with TETRA networks.

Currently there is a rich variety of devices to work with TETRA communications. Major manufacturers (e.g. Motorola, Nokia and Siemens) include TETRA solutions as part of their product lines. TETRA devices cover a range of fixed terminals, mobile handsets, pagers and mobile devices for vehicles. Encompassing the possibilities given by the standard, this equipment is able to deliver services on mobile radio, cellular telephony, text messaging and digital data.

Records presented by Ascom Group indicate that TETRA handset prices show a decreasing trend since 2003, and predictions in the industry foresee a continuous price pressure towards the level of a regular cellular phone (ASCOM, n.d.). Nonetheless, despite of the growing number TETRA products out in the market, pricing and availability do not match those shown by GSM-supporting equipment.

3 MOTIVATON

As said, TETRA product lines are not as wide as equivalent GSM equipment. It is our motivation to take advantage of the versatility and the diversity of cutting-edge GSM mobile devices in market to incorporate them into a TETRA environment. This incorporation may bring another advantages, for instance by accessing a TETRA network from a common mobile device, emergency personnel using general-purpose mobiles may gain autonomy and commonality with other interfaces.

It is desired to develop a logic interface that enables regular mobile devices to directly access a TETRA network. Because of its simplicity and disposition to work at different levels, we have chosen text messaging as the first service to analyze.

To develop the logical interface, we selected Android as preferred framework, considering its flexibility and openness. Within Android, different levels of abstraction are offered, and in particular, we assume that the telephony API already included in Android can be a useful element of abstraction to outline a solution.

The ideal scenario would be that at a lower level, a parallel layer is placed to collaborate with the environment that Android already has to manage the GSM phone signals, in such way to map the TETRA channel signaling through the composition of proper Hayes AT commands.

4 RELATED WORK

Comparisons between TETRA and GSM have been frequently presented in TETRA conferences and summits, as well as position papers and white papers that analyze performance or services provided by each standard. Generally, position papers are issued by the TETRA Memorandum of Understanding (TETRA MoU), and white papers are issued by major manufacturers or service providers. These works agreed in pointing out several mission critical characteristics from TETRA that cannot be found in other similar standards.

We have reviewed as well a number of case studies showing results of the use of TETRA (i.e. performance metrics) in a given environment, and experiences from user organizations using TETRA around the world. As part of their user experiences, several comparisons between TETRA and similar technologies are mentioned too. However, to the extent of our current review, we were not able to find any work whose aim is to incorporate TETRA services to GSM equipment through a customized software layer.

5 EXTENSIONS TO ANDROID TELEPHONY SUPPORT TO HOLD DIVERSE STANDARDS

To establish an outline of the conditions that satisfy the necessities described in section 3, first we need to know the similarities and differences between GSM and TETRA, as the former is currently supported by Android through different management libraries that let users to access different GSM services. We consider that this availability represents an important starting point to set the directions for our work.

Once we know the similarities and differences between the two standards, a general study on the way that Android supports GSM will take place, for us to understand how Android manages GSM telephony, messaging and data services and how the currently offered management may be leveraged or re-used to access TETRA. Given the case that it is not possible to baseline or re-use GSM libraries to fit TETRA operations, other strategies (e.g. kernel programming) may be considered. After that, we will be in a position to draw conclusions which at the same time will serve as foundation for the future work that is needed to incorporate TETRA support to Android devices.

In light of this strategy, the following activities were conducted: 1) Know differences between TETRA and GSM, 2) Understand how SMS is managed by GSM to leverage the required behavior for TETRA-SDS, 3) Determine how Android provides software packages to manage GSM, 4) Review the different abstraction levels that Android has to manage GSM and 5) Select the most ideal level to implement the necessary functionality. The rest of this section is organized matching the order of these activities.

5.1 Differences between TETRA and GSM

ET Industries highlight that TETRA is not intended to compete with GSM or any other cellular technology; they are clearly made for different purposes. While TETRA was designed for PMR applications, GSM was designed for public cellular telephony (ETI, n. d.). Table 1 shows a summary of different mobile services and its availability through our two relevant standards.

Table 1: Comparison of services offered by GSM and TETRA.

Service	GSM	TETRA
Individual call	Supported	Supported
Group call	Not Supported	Supported
Emergency call	Supported	Supported
Short data messages	Supported	Supported
Encryption	Supported	Supported
Authentication	Not Supported	Supported
Data services	Supported	Supported
Out of network coverage	Not Supported	Supported

It has been claimed that TETRA is more robust than GSM (Nordman et al., 2001). TETRA offers some specific features that cannot be found in GSM, including a number of features designed for safety and emergency use. For example, it allows one-to-one and one-to-many communication modes, and thanks to its technology based on a four slot system, it may use multiple carrier frequencies to significantly increase the utilization of the radio channel (ASCOM, n.d.).

Commercial GSM services are not mission-critical; they collapse under heavy load (i.e. during the emergency events of London underground bombing and Madrid railway bombing). In the other hand, TETRA has proven the capacity to handle location updates of thousands of units in dense city areas under highly demanding conditions, like those in the 2008 Olympic Games in Beijing: 86.000 users, (more than 40.000 from Beijing Police), holding a total of 17 million group calls during the whole Games period (Korpilahti, 2010).

Major high-level differences between GSM and TETRA systems may be summarized as follows (ETI, n.d.) (Swan, 2006) (Pasquali, 2007).

Table 2: Summary of differences between GSM and TETRA.

	GSM	TETRA
Purpose	Public cellular telephony	Professional mobile radio applications
Freq. range	900 to 1910 MHz	380 to 430 MHz
Channel access method	Frequency Division Multiplication Access	Time Division Multiplication Access
Call set-up time	Seconds. (Not suitable for emergency services)	Milliseconds. (Suitable for emergency services)
Security	No authentication for listening parties	Authentication plus encryption
Network tolerance	Congestion of in demanding situations	Tolerance to congestion in stressing situations

5.2 SMS Management on GSM

The Short Message Service makes available a means of sending messages of limited size to and from mobile devices. The provision of SMS makes use of a service center, which acts as a store and forward center for short messages.

It is referenced by the 3rd Generation Partnership Project that a GSM SMS message can be up to 160 characters long, where each character is 7 bits according to the 7-bit default alphabet. There are two ways of sending and receiving SMS messages: by text mode and by Protocol Description Unit (PDU) mode. The text mode is an encoding of the bit stream represented by the PDU mode. Both text and PDU modes are all set by the AT-commands when reading the message in a computer application. The PDU string contains the message and meta-information about the sender, service center, time stamp, etc. (3GPP, 2010)

The Short Data Service of TETRA is a data service that is comparable with the SMS of GSM. A TETRA SDS message can carry up to 140 byte data per message. SDS messages may as well be sent and received through text mode and PDU with equivalent AT commands (Ozimek and Gorazd, 2002).

5.3 Software Packages Provided by Android to Support GSM

Android offers APIs for utilizing specific telephony features, (e.g. phone calls and SMS messaging). In the beginning, Android delivered APIs for GSM-specific features, but now these APIs have been deprecated to incorporate new versions that support both GSM and Code Division Multiple Access (CDMA).

Current telephony APIs are structured within a package (android.telephony) and its corresponding classes. These APIs monitor the basic phone information, such as the network type and connection state, and give methods for manipulating phone number strings and other telephony services. For instance, we may find in Android's online reference that it has classes to access to information about the telephony services on the device, or classes to manage SMS operations such as sending data, text SMS messages, and PDU SMS messages (Android, 2011a).

These appealing concepts lead our next steps to research the level in which these APIs are located, and how they interface with the rest of the mobile's environment.

5.4 Abstraction Levels in Android to Manage GSM

Android has different abstraction layers to place a variety of services and applications according to specific requirements and necessities (Santi, Guidi, and Ricci, 2010).

Android's Radio Interface Layer (RIL) provides an abstraction level between Android telephony applications (android.telephony) and radio hardware (see Figure 1). This RIL is radio agnostic, this means that it works with any of the frequencies that the radio hardware can receive. Now Android RIL includes support for GSM-based radios.

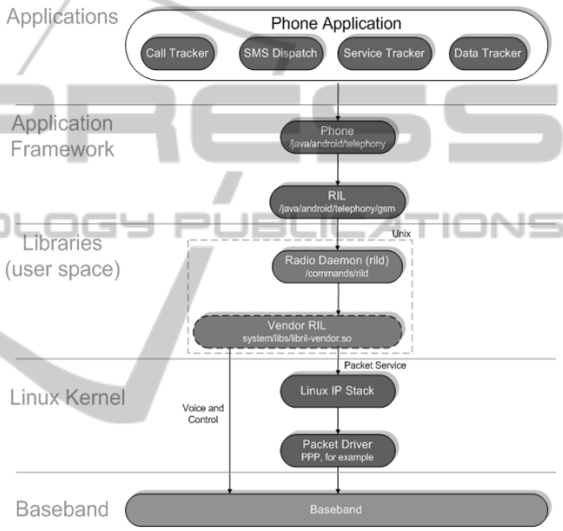

Figure 1: RIL in the context of Android's telephony system architecture (Android, 2011b).

The RIL consists of two primary components: 1) RIL Daemon: initializes the Vendor RIL, processes all communication from Android telephony services, and dispatches calls to the Vendor RIL as solicited commands, and 2) Vendor RIL: a radio-specific vendor RIL that processes all communication with radio hardware and dispatches calls to the RIL Daemon through unsolicited commands (Android, 2011b).

To develop a radio-specific RIL, it is required to create a shared library that implements the set of functions required by Android to process particular radio requests. In an abstraction level, we can locate libraries as software components that rely on the kernel and provide higher level functionality to user-level applications (Ponomarenko and Rubanov, 2010).

5.5 Selection of an Abstraction Level to Implement New Functionality

Since the Android radio interface is radio agnostic, the Vendor RIL can use any protocol to communicate with the radio. Android has a reference Vendor RIL, using the Hayes AT command set, that can be used as a quick start for telephony testing and a guide for commercial vendor RILs (Android, 2011b). This suggests the possibility to take advantage of these libraries to communicate high level APIs with a TETRA radio. Android itself locates the source code for the reference RIL at /commands/reference-ril.

This approach is preferable to writing code in kernel mode, because the adaptation of the physical transport device and a serial signal would need the creation of a driver, whilst an application-level solution would be based only on the adaptation of APIs and the execution of AT commands at user mode.

6 FUTURE WORK

Established the grounds from which this work should be undertaken, the following high-level stages shall be covered to achieve full TETRA functionality on an Android-operated target device:

First, it is necessary to create a shared library that implements the set of functions required by Android to process particular radio requests, in our case, TETRA-specific ones. The required functions are defined in the RIL header (/include/telephony/ril.h), and such library shall be reviewed in sufficient detail to determine what functions are required, how to adapt them and finally how to deploy them in a real-world device. This new RIL should define a handle to the functions that will process all radio requests, and particular functions to handle solicited and unsolicited commands.

The next step is to perform a thorough review on current APIs delivered by Android to suit GSM operations. As mentioned above, our goal is to determine the feasibility and effectiveness to use them to meet TETRA requirements, and then perform the necessary changes or development to deliver a suite of Android APIs that truly allow an Android device to interact with a TETRA network, starting from text messaging (SDS). This stage will represent a major endeavor that will require the design and implementation of efficient methods, able to interact both with the RIL and with high-level user applications. Currently we are reviewing different strategies to achieve this goal in the most appropriate fashion.

Though Android permits full interaction with Java, we acknowledge that other programming languages (e.g. C or C++) facilitate hardware and low-level interfacing. Due to this, it is our goal to explore the possibility of writing or baselining the necessary functions using native C code, considering the possibilities that the Android Native Developer Kit (NDK) gives.

Writing or re-utilizing C code should not introduce conflicts with the rest of the Android environment. Applications written in native code are still packaged into the usual Android .apk file and they still run inside of the virtual machine on the device, leaving the fundamental Android application model untouched (Android, 2011c). The use of Android NDK will enable the development effort to manage the device in an adequate level of detail beyond the limit of the common framework (Sangchul and Jeon, 2010).

Currently, we have exercised different examples on how C code can be written and called by regular Android applications as native methods. Results have been positive in methods that return values, arrays and simple objects. For interaction with complex objects, the use of aid tools like SWIG may be advisable. In the same sense of the performance tests described by Sangchul and Jeon (2010), our observations have been as well substantially favorable to native C code when comparing its execution time with identical methods written in Java.

7 CONCLUSIONS

The main contribution of this paper is to provide a background on how Android framework can be modified in a way to allow it to interact with services from different communication standards, additional to GMS or CDMA. Although significant effort is yet to come, essential information may be taken from this paper to set a projection in the forthcoming work:

- Differences between GSM and TETRA are located chiefly in physical and transport layers. This proposes that software layers may deal with different sorts of packages delivered by radio hardware.
- Android offers different abstraction levels to interface with the context of telephony system architecture, including RIL, a radio agnostic layer that may work with all frequency ranges

that the radio can receive. This RIL interfaces with telephone APIs and libraries within user space. Vendor RIL can be modified or replaced to implement a radio-specific RIL, in this case, a TETRA-specific one.

- Thanks to the RIL abstraction layer, it is visualized that no kernel programming would be required but the design and implementation of a TETRA-specific RIL library at Android's user space level, capable to work with APIs that suit specific needs required by TETRA.
- This recommends in consequence undertaking the effort of creating the RIL that implements the functions required by Android to process TETRA radio requests, and re-using and tailoring android.telephony APIs to satisfy the requirements set by the new RIL library. We foresee the use of Java, as well as the inclusion of native C code as a convenient tool to incorporate functionality related to low-level operations.

Our goal is to provide functionality in fundamental services such as text messaging interchange. At this point it is not yet possible, though, to assure that all functionality and enhanced features provided by TETRA (encryption, direct mode operation, etc.) can be supported only by adapting basic telephone and messaging Android APIs.

REFERENCES

Android 2.3 Release Documentation (2011). *Android Package Summary: android.telephony.* Retrieved February 8, 2011, from http://developer.android.com /reference/packages.html.

Android 2.3 Release Documentation (2011). *Android Radio Layer Interface*. Retrieved February 8th, 2011, from http://source.android.com/porting/telephony.html

Android 2.3 Release Documentation (2011). *Android Native Development Kit*. Retrieved February 8th, 2011, from http://developer.android.com/sdk/ndk.

Ascom Group. (n.d.) *TETRA Terrestrial Trunked Radio*. © Ascom. Retrieved February 8th, 2011 from http:// www.ascom.com/en/tetra-article.pdf.

Axiotis, D. I., Xenikos, D. (2007). On the effects of short data service-transport layer message length in QoS metrics of TETRA networks. *Proceedings of the 18th Annual IEEE International Symposium on Personal, Indoor and Mobile Radio Communications.*

Bakaric, S., Borzic, M., Bratkovic, D., Grga, V. (2005). TETRA (Terrestrial Trunked Radio) - Technical Features and Application of Professional Communication Technologies in Mobile Digital Radio Networks for Special Purpose Services. *Proceedings of the 47th International Symposium ELMAR-2005.*

Bissmeier, R. (2006). High demands on cellular networks-High demands on Surge Protective Devices. *Proceedings of the 28th Annual International Telecommunications Energy Conference.*

ET Industries. (n.d.). *Introduction to TETRA Technology*. Retrieved February 8th, 2011, from http://www.eti world.com/tetra.pdf.

Korpilahti, T. (2010). TETRA Data Applications. *TETRA Moving Forward in Indonesia. Jakarta 2010.* Retrieved February 8th, 2011, from http://www. tetramou.com.

Mikulic, M., Modlic, B. (2008) General System Architecture of TETRA Network for Public Safety Services. *Proceedings of the 50th International Symposium ELMAR-2008.*

Nokia. (2004). *TETRA Quick Guide* © Nokia. Retrieved February 8th, 2011, from http://www.nokia.com /NOKIA_COM_1/About_Nokia/Press/White_Papers/ pdf_files/tetraquickguide.pdf.

Nordman, M. et al. (2001). TETRA radio in Monitoring and Control of Secondary Substations. *Proceedings of the 2001 Developments in Power System Protection Conference.*

Ozimek, I., Gorazd, K. (2002). SCADA System Using TETRA Communication Network. *Proceedings of the 6th WSEAS International Multiconference on Circuits, Systems, Communications and Computers.*

Pasquali, F. (2007). The TETRA Business Case. *TETRA today in Turkey*. Retrieved February 8th, 2011, from http://www.tetramou.com/tetramou.aspx?id=4208 TETRA MoU.

Ponomarenko A., Rubanov. V. (2010). Automated Verification of Shared Libraries for Backward Binary Compatibility. *Second International Conference on Advances in System Testing and Validation Lifecycle.*

Sangchul L., Jeon, J. W. (2010). Evaluating Performance of Android Platform Using Native C for Embedded Systems. *2010 International Conference on Control, Automation and Systems.*

Santi, A., Guidi, M., Ricci, A. (2010). Exploiting Agent-Oriented Programming for Developing Android Applications. *Undicesimo Workshop Nazionale "Dagli Oggetti agli Agenti". WOA 2010.*

Swan, D. (2006). Comparing TETRA with other Technologies. *TETRA Experience Dubai 2006.* Retrieved February 8th, 2011, from http://www. tetramou.com.

TETRA (Terrestrial Trunked Radio) Memorandum of Understanding (2011). Official Website. © 2011 The TETRA MoU Association Ltd. Retrieved February 8th, 2011 from http://www.tetramou.com.

Third Generation Partnership Project. (2010). TS 23.040 V9.3.0: Technical realization of the Short Message Service (SMS) 3GPP. © 3GPP Organizational Partners. 2010.