Taxonomy of Tools and Techniques for Network Monitoring and Quality Assurance in 3G Networks

Shaifali Gupta and Rashi Garg
Indraprastha Institute of Information Technology
Delhi

ABSTRACT

3G cellular networks changed the face of technology completely by bringing to people's doorsteps high speed internet with improved bandwidth and better quality of experience. This accelerated the development of mobile apps harnessing the newly developed trend of near 24X7 connectivity on mobile phones. Today we have numerous applications fitting in the categories of VoIP, chat, social networking and mobile gaming. However, in the hassle of launching another feature rich app which can take the market by storm, its efficacy in terms of resource utilization and impact on the network is one area which often takes the back seat. Developers in majority are ignorant of the intricacies and modus operandi of 3G networks, and shortage of effective tools which can help them visualize the network efficiency of their app further aggravates the problem. The tools and techniques for quantifying an app's behavior in terms of its network efficiency or for the general purposes of network monitoring and bench- marking, are not only limited in number but they often go unnoticed as well. In this work, we target exactly that issue. We provide taxonomy of available tools and techniques for network monitoring and performance measurement in 3G networks. We try to categorize the tools on the basis of their functionality and use-case. Key features of every tool and technique are also highlighted. In the end, we present in tabular form important features used for classification and tools that incorporate them.

Keywords

3G Networks, Quality Assurance, Network Monitoring

1. INTRODUCTION

3rd generation mobile network, commonly referred to as 3G, witnessed a rapid adoption by telecoms and general public alike and has now become all pervasive. This coupled with the advent of smart phones lead to an upsurge in the usage of mobile data accompanied with the development of a plethora of mobile apps. As a result, a major part of mobile traffic is now data, not voice. However, there have been little to no changes in the existing infrastructure to support the increased demand from both voice and data perspective. The traditional infrastructure which was initially designed keeping in view only the voice traffic, is now being used to handle the data as well. This has given rise to new challenges for the network operators and generated a newer set of vulnerabilities in the system. Universal mobile Telecommunications System (UMTS) is by far the most popular 3G technology. It uses Wideband Code division multiple access (W-CDMA) as its air interface standard. The radio access network for UMTS, generally called UTRAN (Universal terrestrial radio access network) is made up of Node B's and Radio Network Controllers (RNCs). A single RNC handles a large number of Node B's. This places the bulk of load of message processing on RNCs.

Radio resource control layer (RRC) protocol [3] is responsible for managing the control plane traffic between the user equipment (UE) and base station. RRC protocol is deterministic in nature and contains three main states - IDLE, FACH and DCH (See Figure1). When the UE is not connected to the network, it is by default in IDLE state. Any net- work activity from the UE (reception or transmission of data packets) causes it to move to one of the connected states - FACH or DCH. FACH is chosen for lower buffer occupancies and provides lesser throughput than DCH. DCH sets up a dedicated channel for data transfer between the UE and base station. It is also the highest energy consuming state and causes significant battery drain at the UE. However, it provides better throughput than FACH. Several factors can trigger a state demotion from DCH to FACH/IDLE or from FACH to IDLE. A transition from DCH to FACH can occur due to lower buffer occupancy. Another reason can be expiry of inactivity timer which checks against prolonged inactivity at a higher energy consuming state. The timer values are fixed by operators and are not exposed publicly. Each data activity resets the timer value. However if the period of inactivity after the last data transmission/reception crosses the threshold set by the operators, the timer expires and the UE is demoted to a lower energy consuming state, which is FACH in this case. A similar timer guards the activity of the UE for FACH to IDLE transition.

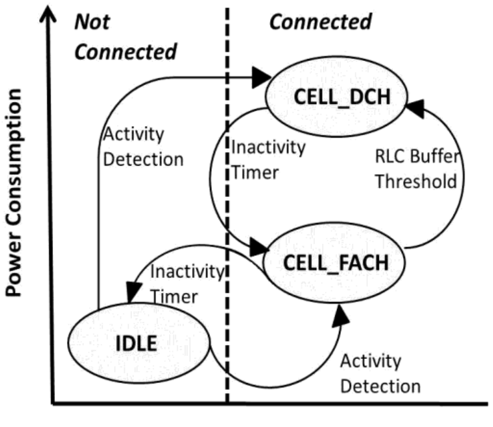

Fig. 1: RRC State Machine (From: Android Phone Based Appraisal of App Behavior on Cell Networks [1], p.2)

2. MOBILE APPS

A wide variety of mobile apps offering different functionalities are present at user's disposal these days. While the application developers are only concerned about producing a feature rich app which goes popular immediately, network operators are faced with the challenge of maintaining the quality of experience for the user without requiring a significant overhaul in their existing infrastructure. This conflict of interest between the developers and operators is aggravated due to the inability of developers in gauging the impact of their application on the network. In most of the cases, an application developer has no visibility of the control plane signaling caused due to his application and its effect on the network. This leads to the development of badly designed applications which cause significant wastage of radio resources and deteriorate the network performance. For an end user, the experience can be in form of frequent disconnections, battery drain and poor network connectivity. More problematic are the applications which stay active even in the background without any user interaction. These applications, for example viber, skype, whatsapp etc. send intermittent short bursts of data in the background causing the radio resources to get allocated and deallocated frequently. This chatty behavior generates significant control plane messages, thereby overloading the already constrained radio resources [16]. There have been certain studies [14], [13], [10], [16], [22], [2] on the use case and network impact of some popular mobile applications. Different methodologies have been adopted for defining and measuring the efficiency of an app under study. The techniques range from mathematical modeling for inference of RRC state machine behavior to development of tools for capturing the network activity or RRC state transitions caused by an app. Some researches [2], [22] dived deep in to the problem and tried to capture the interplay of application activity and control plane signaling in different scenarios. In addition to this, there also exist proprietary tools and solutions for network performance measurement, monitoring, refactoring, and capturing of low level control plane messages. These, along with their open-source counter- parts add significantly to the breadth of tools available for 3G landscape.

This work attempts to investigate state of the art tools and techniques for measuring/monitoring application behaviors and network performance for 3G networks. We have tried to develop a taxonomy of tools by covering almost all popular techniques at the time of this writing. We present a brief description of the tool/technique followed by its key features and possible limitations. We classify the techniques based on their use case and functionality. In the end, we tabulate all the tools and mark the standard features they cover.

3. OVERVIEW OF TOOLS

In order to measure the impact of activity of an application on the network, it is necessary to accurately determine the network traffic generated by the application. On the side of networks, one has to monitor the behavior of RRC state machine for quantifying the network's performance. An accurate correlation of these two factors can give useful insights on how efficiently the radio resources are utilized by the UE (specifically by the apps running on the UE). Thus the problem of inefficient utilization of network resources can essentially be broken down into two sub-problems-

- Capturing the network activity of an application
- Detecting the behavior of RRC state machine

There exist specialized tools and techniques for solving each of the above mentioned problems individually. Fortunately, we also have more complete packages offering a comprehensive solution. Thus, we can add one more category of tools which target both of the above mentioned problems simultaneously. In this work, we refer them as- 'All-in-one Tools'. There also exist specialized tools to process tcpdump output and provide detailed analysis and insights on the captured packet dump. This gives us one more category of tools, specifically for 'Data Analysis'. Thus we have two more categories-

- All-in-one Tools
- Tools for data analysis

3.1 Capturing the network activity of an application

In this section, we list the tools which can be used for capturing network packets along with their brief description.

3.1.1 Tcpdump

Tcpdump [19] is a popular tool for network monitoring and packet capture. It uses libpcap [11] for its functioning, which is a C/C++ based packet capture library. The tool does not have any Graphical interface and works only from command line. Originally, the tool is built to work on unix like systems, but we also have its Windows counterpart - Windump for people with windows distribution. A number of optional features are also available with the command-line interface which can be enabled using corresponding flags. Once it finishes capturing the packets, it outputs count of packets that were captured, packets that matched the filter supplied through the command-line (output for this varies depending on the operating system) and packets dropped by kernel. The tool provides powerful packet filtering and analysis capabilities. Packets can be saved into a file with .pcap format, which can later be opened for analysis by any compatible software (like wireshark).

Key Points

- Open-source
- Command-line interface
- Lightweight
- Powerful packet capturing and filtering capabilities
- Can run concurrently with other softwares.
- Supported on Unix as well as Windows.

3.1.2 Network Log

Network Log [8] is an open-source application for rooted Android phones. It monitors iptables-logging to determine which apps are making network connections. It then pro- vides detailed statistics about each such connection including source and destination addresses, network protocols, number of bytes transferred and timestamp. While tcpdump gives a consolidated view of packet exchange without distinguishing between the applications generating those packets, Network log gives a detailed application level mapping. It provides real-time network usage statistics along with an option to view overall timeline for all apps.

Key Points

- Open-source [9]
- Works only on rooted android devices.
- Provides application level mapping.
- Real time logging.

3.1.3 Snoop

Snoop [18] is a built-in sniffer that comes with Solaris operating system. It does not have any graphical interface and works only through command-line. By default, it puts the device in promiscuous mode where it captures all packets in the network. This mode can be switched off using corresponding flags. Along with an option to view data in real-time, it also allows data to be saved in a file for offline viewing and analysis. The file format for snoop is different from tcpdump. However, inter-conversion is possible with Etheral tool 'editcap'. By default it captures both

IPV4 and IPV6 packets. The snoop command can be run only in super-user mode.

Key Points

- Available only for solaris operating system.
- Superuser privilages required to run the command.
- No graphical interface.
- File format different from Wireshark.

3.1.4 Shark for Root

Shark for Root is a packet capture application developed for rooted android phones. It requires Tcpdump binary on the phone in order to work. The packets are saved in a file in pcap format and can be viewed offline using wireshark or any other application that can read pcap files. Since the application is a wrapper on Tcpdump, it does not offer any significant benefit over using standalone tcpdump binary other than a fancy user interface.

Key Points

- Built for Android
- File saved only in pcap format
- Uses Tcpdump behind the scenes, so no visible advan- tage over tcpdump.

3.1.5 Microsoft Network Monitor

Network Monitor [5] is a protocol analyzer and network capture tool by Microsoft. It is available only for Windows platform. The tool can be downloaded for free from its official page. It offers powerful packet filtering capabilities which can be applied at the time of capture or for display only. NMCap is its command line counterpart. It also provides API (NMAPI) which allows to access parsing and capturing engine programmatically. With the help of this API, developers can extend the capabilities of the tool beyond vanilla capturing and filtering.

Key Points

- Free Download.
- Both graphical and command-line facilities.
- Compatible with Windows operating system only.
- Light weight.
- API to enable the developers to extend the capabilities of the tool.

3.2 Tools for Data Analysis

These tools are mainly for processing logs of captured packet information, sanitizing data and generating statistics. Though some of them can also provide packet capture functionality, the main selling point of these tools is their analysis capabilities.

3.2.1 Tcptrace

Tcptrace [20] is an extremely powerful tool for processing tcpdump output files. It is strictly an analysis tool. In addition to tcpdump, it can also process files produced by other packet cpature utilities like snoop and Windump. It produces out- puts in different formats each containing the connection seen, bytes tranferred, round trip times, throughput and more. While the tool is mostly used as a commandline utility, it can also produce results in graph format using the X Window program xplot. The graphs include throughput graphs, time sequence graphs and graphs of round trip times. This tool is more useful for analysis of network performance and related problems than for inspection of individual packets.

Key Points

- Open-source.
- Works on Unix, Windows and Solaris systems
- Strictly analysis tool. No packet-capture capabilities.
- More useful for network troubleshooting than individual packet analysis.

3.2.2 Wireshark

Wireshark [23] is a powerful packet capture and analysis tool that supports multiple operating systems including Windows, Linux, Solaris, NetBSD etc. It provides deep packet inspection for numerous protocols and excellent packet filtering capabilities. The tool has an easy to use graphical interface. It provides capabilities of live capture along with an option of loading logs from previously saved files. The files can be saved in multiple formats including pcap, microsoft network monitor, Cisco Secure IDS iplog etc. Tool is not built to run on mobiles. It can be freely downloaded for use from its official site.

Key Points

- Free Download
- Excellent Graphical Interface
- Supports multiple platforms
- Support for multiple network technologies including ethernet, wifi, wimax, ATM, Bluetooth, USB, Token- ring etc.
- Deep packet inspection for number of protocols.

3.2.3 Microsoft Message Analyzer

Microsoft Message Analyzer [4] is a successor to Microsoft Network monitor with new features and extended capabilities. However, the glaring difference between Microsoft message Analyzer and its predecessor -Network Monitor, qualifies it for a separate mention. It is much more than a simple packet capture and analysis tool for Windows operating system. It can capture live events at various system and endpoint levels. Parsing and validation of protocol messages and sequences is another key feature. Its excellent display features enable the user to view trace, log or any other message data in multiple formats like tree grid view, interactive tool windows and other customizable views employing charts, graphs and timeline components.

Key Points

- Successor to Microsoft Network Monitor.
- Works only on Windows platform
- Powerful display features.
- Free Download available.

3.3 Detecting the behavior of RRC state machine

Detection of RRC state machine behavior is difficult in absence of support from network operators. Network operators do not share these details willingly, as it can directly affect their market competence. Hence, one has to rely on costly commercial solutions which are often limited by selective hardware support. This section explores some popular commercial tools and other viable alternatives for detection of RRC state machine parameters.

3.3.1 *Secret codes*

Secret codes [17] are manufacturer specific alphanumeric codes which can be dialed from the phone. They are mainly used by network and radio engineers for debugging, troubleshooting and resetting different network specific parameters. There exists a wide variety of such codes for different manufacturers and operating systems. However, not all of them cater to the specific purpose of detecting 3G network specific parameters and in particular, the RRC states. The only code available for detecting RRC state of an android phone at the time of this writing is: *#0011#. As stated earlier, these codes are not universal, but are specific to certain devices/ manufactures/ chipsets and operating system of the phone. The above mentioned code is only supported by Sam- sung phones. Two other codes which might find use in this problem setting are: *#9900# and *#*#4696#*#*. While the first one (*#9900#) can be used for enabling/disabling fast dormancy on the phone, the latter allows the user to select a particular connection mode to the network (WCDMA only, GSM, WCDMA preferred etc.).

Key Points

- Supported only on selected devices.
- Process cannot be made to run in background.
- Default serviceMode.apk cannot be interfaced with custom programs to fetch the states programmatically. No known solution for interfacing at the time of this writing.
- States are actual, not inferred.
- Codes are easily available on internet.

3.3.2 *QXDM*

Qualcomm Extensible Diagnostic Monitor (QXDM Professional) [15] is a proprietary tool owned and distributed by Qualcomm Technologies. It can provide extensive real time logging of layer 3 control plane signaling messages exchanged by the phone. This tool runs outside the phone, preferably on a PC running on Windows XP/ 2000 and supports phones with a Qualcomm chipset only.

Key Points

- Commercial Tool.
- Can capture actual layer-3 control messages exchanged by the device.
- Works on selected devices. Only on those with a Qualcomm chipset.
- Support for multiple standards like WCDMA, EvDO, CDMA2000 etc.

3.3.3 *3G3T*

3G transition triggering tool (3G3T) [12] is a python based active measurement tool which allows the user to measure various RRC state transition parameters including the state inactivity timers and enables determining delay of different RRC procedures like channel set-up and paging. 3G3T typically runs on a laptop equipped with a 3G data card and a fast Ethernet card. Thus the laptop is simultaneously connected to the 3G network under test and to the public internet. For performing various measurement tasks, 3G3T sends data from one interface and receives traffic at other interface, simultaneously recording different parameters like Inter-departure time of packets, one way delay, timestamps etc. Packets sent and received at both the interfaces are logged. RRC states are inferred and timer values are calculated by performing different experiments using the above mentioned setup. In the paper describing 3G3T [12], authors have validated their findings with Nemo outdoor.

Key Points

- Supported on Windows and Linux.
- Test device should have at least one more network interface other than cellular network interface.
- Since the experiments are conducted on a single device, expensive clock synchronization is not required.
- RRC states are inferred through experiments. No actual control plane messages are captured.
- Not built to run on mobile devices.

3.4 All-in-one Tools

In this section, we describe tools which offer a more complete and comprehensive solution. All the tools mentioned below provide the functionalities of both RRC state detection and network packet capture.

3.4.1 ARO

ARO [14] is a cross layer analysis tool which provides visibility of control plane traffic to the end user. In addition to this, it also performs an in-depth analysis for TCP and HTTP layer and provides a process to packet mapping. The tool consists of two parts- an "on the phone" data collector and an offline analyzer. Data collector is built by extending the functionality of Tcpdump to include capture of user initiated events (pressing of buttons and touching the screen etc.) and a packet to application correspondence. Analyzer is implemented in C++ and runs in offline mode on a commodity PC. It takes as input the packet traces captured by the online data collector to perform a series of different analyses. In particular, it performs RRC, TCP, HTTP, and burst analysis. For RRC analysis, the analyzer uses predefined handset and carrier specific configuration parameters to simulate RRC state transition machine. In simpler words, RRC state inference algorithm used by ARO takes the packet traces and some predefined parameters to estimate or infer the RRC states of the device at various stages of capture. Thus, depending on the correctness of the algorithm there is a risk that the inferred RRC state may differ from the "actual" state of the device. Developers of this tool validated its correctness by comparing it against power consumption based inference approach which relies on the assumption that there is a significant difference in power consumption of handset during different RRC states.

Key points

- Open-source.
- Data collector part works only on rooted android devices.
- Analysis is done in offline mode on a commodity PC.
- Uses mathematical models to infer the RRC states and corresponding state machine parameters.
- Extends the capabilities of tcpdump.

3.4.2R*ILAnalyzer*

RILAnalyzer [22] is an open-source cross layer analysis tool for rooted android handsets which enables the user to capture “actual” control plane events, in particular the RRC states along with an accurate logging of user plane events. The tool currently supports only those android devices which use an Intel/Infineon XGold Chipset. For control plane logging, they replace the phone’s inbuilt ServiceMode apk with their own system tool which runs in the background at polls the phone’s modem at maximum frequency the device responds. For user plane logging, they extend NetworkLog which uses IPtables with Nflog for packet logging and packet to process mapping.

Key
Points

- Open-source.
- Extends Networklog for packet capture.
- Replaces servicemode apk of the device for detection of actual RRC states, as seen by firmware on device’s modem.
- Works on rooted android devices.
- Supports only selected handsets.

3.4.3*TEMS Investigation*

Tems Investigation [21] is another popular proprietary tool for performing measurements, verification, optimization and maintenance of wireless networks. TEMS Investigation provides a complete solution to network operators by offering capabilities of data collection, real time analysis and post processing. TEMS Investigation supports nearly all cellular network technologies including but not limited to GSM/GPRS/EGPRS, WCDMA/HSPA/HSPA+, LTE, CDMA-2000/CDMAone/Ev-Do, WiMax etc. The application runs on any windows based PC/laptop. For data collection, it interfaces with phones, scanners and other data collection devices and logs the data for further processing. It supports user terminals from multiple vendors.

Few examples are – LG, Sony Ericsson, HTC, Nokia, Huawei, Samsung etc. It provides capabilities for different measurement tasks including logging of lower layer control packets.

Key Points

- Commercial.
- Can run only on Windows based systems.
- Interfaces with user terminals for data collection and logging control packets.
- Supports multiple network technologies.

3.4.4 Nemo Tools Package

Anite plc provides an array of tools for performing various network measurements, troubleshooting, benchmarking and to serve different needs of network drive testing. Nemo Outdoor [6] and Nemo Handy [7] are the main tools falling in this category.

Nemo Outdoor: Nemo Outdoor is a powerful engineering tool for measuring air interface of wireless networks. It supports several networks and technologies includ- ing but not limited to – GSM, GPRS, CDMA2000, WCDMA, EDGE, HSDPA and WiMax. Measurement results generated by Nemo outdoor are useful for net- work planning, rollout, optimization, verification and maintenance. Measurement set-up for Nemo Outdoor consists of a PC or Tablet PC with Windows operating system and Nemo Outdoor software, Nemo outdoor compatible test terminals (mobile phones or scanning receivers), and a GPS receiver with antennas. Through a customizable interface, user can enable logging of IP packets which are saved in a format compatible with Wireshark software. Layer 3 control messages including the RRC states can also be logged in case of UMTS WCDMA networks. The log files can later be pro- cessed using Nemo Analyze , a strong post-processing tool developed by Anite, or any third party vendors.

Nemo Handy: Nemo handy is a handheld device which can be used to perform extensive network measure- ment tasks along with application QoS/QoE testing while being simultaneously used as a regular phone. Nemo Handy family consists of Symbian based Nemo Handy –S and Android based Nemo Handy – A. They offer smart solutions for air interface measurement of EDGE/WCDMA/HSDPA/HSUPA and many other wireless networks. Most network parameters including layer-3 signaling messages are logged and made avail- able for post processing using Nemo Outdoor/ Nemo Analyze or third party vendors supporting the output file format.

Key Points

- Commercial
- Nemo Outdoor software requires Windows based PC and test terminals.
- Can do accurate logging of layer 3 control messages along with capture of IP packets.
- Nemo Handy is built for mobile devices and supports symbian and android operating systems.
- Nemo Handy does not work on any commercially available commodity phone, but provides its own customized devices with the purchase of software.

4. CONCLUSION

In this work, we provide a detailed classification of tools and techniques for network monitoring and quality assurance in 3G networks. We believe this would help researchers, network operators and other users to select appropriate tool according to their requirements and usage scenario. We tried to cover all popular tools in every category and described the distinguishing features of each and every tool. We noticed that, while a sufficient number of tools are available for net- work packet capture even for different platforms and target devices, there is a lack of good and affordable techniques for RRC state detection. Number of tools for RRC state detection is so less that it can be counted on fingers. There also exist other constraints like heavy dependency on hardware or requirement of a commercial license. Majority of tools are 'inference based' and may not provide accurate results always. Hence, this is an area which requires significant effort from the research community. The 'All-in-one' tools are a cut above the rest in terms of the range of functionality they provide and comprehensiveness of solution they offer. In the end, we conclude by saying that while 3G networks have evolved a lot, tools and techniques to monitor these networks and to provide quality assurance still have a long way to go.

Table 1: Key Features of different 3G tools

Tool	Open-Source	Superuser privileges	User-plane		Control-Plane		Analysis	N/W	Devices	O.S
			On-phone	Process to packet mapping	On-Phone	Inferred or Actual				
TCP-Dump	✓	✓	✓	✗	N/A		✗	All	All	Linux and Windows
Network Log	✓	✓	✓	✓	N/A		✗	All	All	Android
Shark for Root	✗	✓	✓	✗	N/A		✗	All	All	Android
Snoop	✗	✓	No	✗	N/A		✗	All	All	Solaris
Microsoft Network monitor	✗	✓	No	✗	N/A		✗	All	All	Windows
Microsoft message-Analyzer	✗	✓	No	✗	N/A		✓	All	All	Windows
Tcptrace	✓	✗	N/A		N/A		✓	All	All	Unix, Windows, Solaris
Wire-Shark	✓	✗	No	✗	N/A		✗	All	All	Windows and Linux
ARO	✓	✓	✓	✓	No	Inferred	✓	Mostly	All	Android
Secret Codes	✗	✗	N/A		✓	Actual	✗	Mostly	Selected	Android
QXDM	✗	✗	N/A		No	Actual	✗	Mostly	Selected	Windows
3G3T	✗	✗	No	✗	No	Inferred	✓	Selected	Selected	Any supporting python
RIL-Analyzer	✓	✓	✓	✓	✓	Actual	✓	All	Selected	Android
TEMS Investigation	✗	✗	No	✗	No	Actual	✓	All	Selected	Windows
Nemo Outdoor	✗	✗	No	✗	No	Actual	✓	Mostly	Selected	Winodws
Nemo Handy	✗	✗	✓	✗	✓	Actual	✓	Mostly	Selected	Android, Symbian

Tool	**Open-Source**	**Superuser privileges**	**User-plane**		**Control-Plane**		**Analysis**	**N/W**	**Devices**	**O.S**
			On-phone	**Process to packet mapping**	**On-Phone**	**Inferred or Actual**				
TCP-Dump	✓	✓	✓	✗	N/A		✗	All	All	Linux and Windows
Network Log	✓	✓	✓	✓	N/A		✗	All	All	Android
Shark for Root	✗	✓	✓	✗	N/A		✗	All	All	Android
Snoop	✗	✓	No	✗	N/A		✗	All	All	Solaris
Microsoft Network monitor	✗	✓	No	✗	N/A		✗	All	All	Windows
Microsoft message-Analyzer	✗	✓	No	✗	N/A		✓	All	All	Windows
Tcptrace	✓	✗	N/A		N/A		✓	All	All	Unix, Windows, Solaris
Wire-Shark	✓	✗	No	✗	N/A		✗	All	All	Windows and Linux
ARO	✓	✓	✓	✓	No	Inferred	✓	Mostly	All	Android
Secret Codes	✗	✗	N/A		✓	Actual	✗	Mostly	Selected	Android
QXDM	✗	✗	N/A		No	Actual	✗	Mostly	Selected	Windows
3G3T	✗	✗	No	✗	No	Inferred	✓	Selected	Selected	Any supporting python
RIL-Analyzer	✓	✓	✓	✓	✓	Actual	✓	All	Selected	Android
TEMS Investigation	✗	✗	No	✗	No	Actual	✓	All	Selected	Windows
Nemo Outdoor	✗	✗	No	✗	No	Actual	✓	Mostly	Selected	Winodws
Nemo Handy	✗	✗	✓	✗	✓	Actual	✓	Mostly	Selected	Android, Symbian

5. REFERENCES

[1] S. Gupta, R. Garg, N. Jain, V. Naik, and S. Kaul. Android phone based appraisal of app behavior on cell networks. Technical Report IIITD-TR-2013-003, IIIT-Delhi, October 2013.

[2] Alcatel-Lucent Mobile Application Rankings Report, April, 2014.

[3] H. Holma and A.Toskala. WCDMA for UMTS: Radio Access for Third Generation Mobile Communications. John Wiley and Sons Inc., New York,USA, 2004.

[4] Microsoft message analyzer official page. http://blogs.technet.com/b/messageanalyzer/.

[5] Network monitor official blog. http://blogs.technet.com/b/netmon/.

[6] Nemo outdoor. http://www.anite.com/businesses/netw ork-testing/products/nemo-outdoor.

[7] Nemo handy official page. http://www.anite.com/businesses/network-testing/products/nemo-handy-world

[8] Networklog on google play. https://play.google.com/store/apps/details?id=com.googlecode.networklog.

[9] Github networklog. https://code.google.com/p/iptableslog/.

[10] M. Paolini and S. Fili. The taming of the app. Sponsored By:Seven Networks(http://www.seven.com/), 2013.

[11] Pcap documentation. http://www.manpagez.com.

[12] P. H. J. Per¨al¨a, A. Barbuzzi, G. Boggia, and K. Pentikousis. Theory and practice of rrc state transitions in umts networks. 5TH IEEE Broadband Wireless Access Workshop, July 2009.

[13] F. Qian, Z. Wang, A. Gerber, Z. M. Mao, S. Sen, and O. Spatscheck. Characterizing radio resource allocation for 3g networks. ACM SIGCOMM, November 2010.

[14] F. Qian, Z. Wang, A. Gerber, Z. M. Mao, S. Sen, and O. Spatscheck. Profiling resource usage for mobile applications: a cross-layer approach. ACM Mobisys, July 2011.

[15] Qualcomm extensible diagnostic monitor. http://www.qualcomm.com/media/documents/qxdm-professional-qualcomm-extensible-diagnostic-monitor.

[16] C. Schwartz, T. Hossfeld, F. Lehrieder, and P. Tran-Gia. Angry apps: The impact of network timer selection on power consumption, signalling load, and web qoe. Journal of Computer Networks and Communications, 2013(176217), February 2013.

[17] Android secret codes. http://www.digipassion.com/2012/11/samsung-android-mobilephone-secret-codes.html.

[18] Snoop documentation. http://www.softpanorama.org/Net/Sniffers/snoop.shtml.

[19] Tcpdump documentation. http://www.tcpdump.org.

[20] Tcptrace official page. http://www.tcptrace.org/.

[21] Tems investigation official page. http://www.ascom.com/nt/en/index-nt/tems- products-3/tems-investigation-5.htm#overview.

[22] N. Vallina-Rodriguez, A. Aucinas, M. Almeida, Y. Grunenberger, K. Papagiannaki, and J. Crowcroft. Rilanalyzer: a comprehensive 3g monitor on your phone. ACM Internet Measurement Conference (IMC 2013), October 2013.

[23] Wireshark official page. http://www.wireshark.org/.