TraSt: A Traffic Statistic Framework for Evaluating 3G Charging System and Smart Phone Applications*

Wenshan Fu, Tong Zhao, Kaigui Bian, and Wei Yan

School of Electronics Engineering and Computer Science
Peking University, Beijing, P.R. China, 100871

Abstract. We designed and implemented a traffic analyze framework TraSt (Traffic Statistics) based on tcpdump for Android. This framework can get more detailed traffic usage than any other existing traffic monitor frameworks or applications on smart phones, and help us to explore four aspects of Chinese three major operators' charging system as well as four kinds of popular smart phone applications. Our work helps 3G users and software developers have a clearer understanding of the similarities and differences among the operators' charging services, so as to promote the improvement of software development and operators' services, and our analysis with recommendations to application developers can lead to better system design and network infrastructure support.

Keywords: Framework, 3G, Charging, Smart phone, Application.

1 Introduction

With the rapid development of 3G technologies and smart phones, millions of mobile applications emerge on the market. Report by the Ministry of Industry and Information Technology [4] shows that by the end of 2012, the number of 3G users in China is over 210 million. And the the usage trend will be accelerated with the explosive growth of smart phones in the coming years. Which makes traffic, revenues and profits continue to grow.

However, behind the flourishing 3G and smart phone market, there are also some issues. For the operators' charging system, because they are not open to public, some unreasonable phenomenon is difficult to get a satisfactory explanation. For example, a user only download 1KB size of file, but the charging result will always larger than 1KB. The user also can't know the smallest units of charging: if he only used a small amount of traffic, such as 50B, will it not be charging or charging for 50B or even 1KB? Furthermore, whether some protocol overhead is not taking into account? Etc. For the smart phone applications the lack of rigorous review mechanism in unofficial app markets makes users not

* This work is supported in part by 973 Program of the Chinese Ministry of Science and Technology (No. 2009CB320504) and National Natural Science Foundation of China (No. 61073155 and No. 61001075).

J. Su et al. (Eds.): ICoC 2013, CCIS 401, pp. 206–217, 2013.

sure whether network flows consumed by these applications are really what they need, and among the same kind of applications, which is more provincial traffic?

Existing researches on 3G network and smart phone applications are always focus on common metrics in the general wireless network. Netdiff [12] is a benchmark used to compare different Internet Service Providers (ISP) performance. M. C. Chan et. al. [11] proposed an algorithm to evaluate 3G networks TCP/IP transmission rate and delay. H. Junxian et. al. [9] compared the performance of different smart phone applications under different operator networks. But there are almost no efforts on the study of operators charging system, neither comparison among same kind of smart phone applications.

1.1 Our Contribution

The main contributions of our work are:

1. We have designed and implemented an expandable phone side evaluation framework TraSt (Traffic Statistics). This framework can get more detailed traffic usage than any other existing traffic monitor frameworks or applications on Android. And since it is running on the phone, the result can better reflect users real experience.
2. We have proposed some methods to utilize TraSt to explore four aspects of the Chinese three major operators' charging systems (see section IV.A), and get the following result: a. Operators are charging for the traffic of network layer. b. Some protocol overheads such as ARP and ICMPv6 are not taking into account. c. Different operators have different minimum charging units, B or KB. d. the phone will not receive any packet before it initiate a request. These result will lead 3G users to a clearer understanding of the similarities and differences among different operators' charging services.
3. We also have used TraSt to evaluate four kinds of popular smart phone applications (browser, map, online music player and video call). We made experiments in different scenarios, and accurate comparisons of network traffic between different mobile applications. The results show that even among same kind of applications existing great differences. Our analysis with recommendations to software developers can lead to better system design and network infrastructure support.

The rest of the paper is organized as follows. Section 2 introduces data charging process of 3G networks and smart phone applications. Section 3 presents an overview of TraSt. Section 4 proposes the issues and study methodology. Section 5 shows the result and Section 6 draws the conclusion and further discusses future work. Since Android is the most widely used smart phone operating system [1], the ensuing discussions are all made on the Android.

2 Preliminaries

We first give a brief overview of 3G charging architecture for data services in context of UMTS [8]. Then introduce the classification of smart phone applications

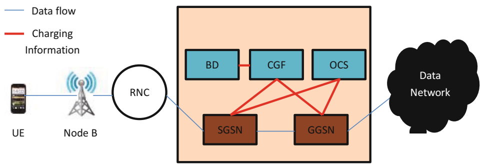

Fig. 1. 3G network data charging architecture

and give the definition of content controllable applications and the reason why we chose them to evaluate.

2.1 3G Data Charging

The overall 3G charging architecture for data services in context of UMTS is shown in Figure 1. 3G network has two main components—the Core Network (CN) and the Access Network (AN) [10]. The AN includes the User Equipment (UE), the Node B, and the Radio Network Controller (RNC). Its main functions are to control access and exchange data with the Packet-Switch (PS) core network. While the PS core network include the Serving GPRS Support Node (SGSN), Gateway GPRS Support Node (GGSN), Charging Gateway Function (CGF), Billing Domain (BD) and Online Charging System (OCS). Its main functions are to delivery data between UE and the external data networks, perform user authentication and charging functions.

When an application needs to access the Internet, the SGSN will perform the authentication function then the UE can access the network through the AN and establish a bearer. For offline charging, whenever a billing event (such as the use of traffic) occurs, the SGSN and GGSN will collect charging information and transfer it to the BD. And for online charging, the charging information will be directly transferred to the OCS, and the OCS will calculate the fee and deduct it from the users prepaid expenses. For more details, please refer to [7].

2.2 Smart Phone Applications

A variety of smart phone applications are now available. According to Google Play, Android applications can be divided into the following categories: 1. Game (Angry Birds); 2. Social(renren); 3. On the go(Google Map); 4. Music and Photos(Duomi Music); 5 Entertainment(Mobile TV); 6. Life(Taobao); 7. Office(Gmail).

For some different applications from a same kind, if we can make the content of traffic basically the same in repeated experiments by certain experimental design and control means, we refer them to *content controllable applications*. For example, we can control the contents of network data of web browser by control its access URL. On the contrary, for some kinds of application such as mail client, different client has different number of messages and information, which we can not control it. We refer them to *content uncontrollable applications*.

For content uncontrollable applications, their data traffic consumptions are largely depend on the actual contents they have accessed, we can not control it, leading the difficulty of comparative experiment control, and making the result unfair. So in this paper, we choose four kinds of content controllable applications including web browsers, maps, online music players and video calls to evaluate.

3 Evaluation Framework

We need a framework to help us explore the 3G charging system and evaluate smart phone applications. In order to meet all kinds of requirements including future needs, this framework should be running on smart phone and be able to record every detail information including protocol header of each layer, the content and the time of send/receive for each packet. As far as we know, there is no such framework to accomplish these tasks. However, this framework is very necessary. since compared with laptops and netbooks, the transmission power and signal strength of smart phones is quite different (generally weaker), a framework can accurately reflect the actual experience of the user only when it is running on the phone. It can bring considerable convenience for the test and evaluation process when it can get detail information of each packet. And it can also reduce manual operation thus making the test and evaluation process more efficient.

Next, we will introduce TraSt, a traffic statistics framework that can meets the requirements. It is an open application framework. Which can get detailed information of each packet, and provide interfaces for the test and evaluation process.

3.1 Challenges

It is difficult to implement such a framework. Although Android is an open source platform and its kernel is Linux, a lot of tools that can be used on Linux cannot be directly ported to Android because of the difference of system architecture. The statistical result should be displayed on UI, so we should implement the framework at the Application Layer. Application Layer's program can only get the System Permissions (lower than the Root Privileges). But the Root Privileges is needed to capture packets. So the best solution is a combination of the Application Layer and Libraries or Linux Kernel Layer. This also involves the Android NDK development [2], which needs cross compiling.

3.2 Framework

The framework of TraSt is depicted in Figure 2. We compiled a Tcpdump [5] which is able to run on Android. And cross compiled the C part of Jnetpcap [3] and libpcap [5] source code with NDK to build a shared library libjnetpcap.so. This shared library will run on Android Libraries Layer. When TraSt starts running, it will call Tcpdump to capture packet and save it to a .pcap file. Then the libjnetpcap.so will read the .pcap file, get the detailed information of each packet and send them to the Application Layer for further processing. The

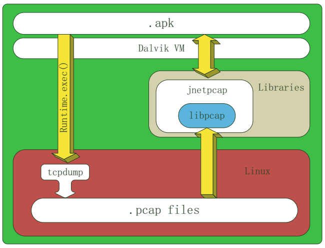

Fig. 2. Framework of TraSt

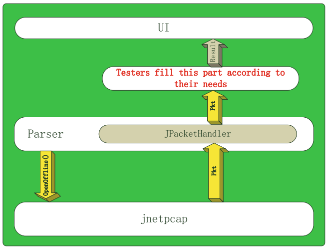

Fig. 3. Framework of Application Layer's apk

Tcpdump, libjnetpcap.so and the Application Layers apk will be automatically deployed to the phone at the time of installation.

The Application Layer code is not fixed, but filled by testers according to their needs, such as get the total data traffic, or average handshake time of Tcp, etc. Figure 3 is the framework of the Application Layer's apk. The major components are the Java part of the jnetpcap source code and the Parser class. The Parser class will get every packet from .pcap file through jnetpcap and callback the nextPacket method of JPacketHandler. What testers need to do is just to implement this method, get what they want, and display the result on UI.

3.3 Scalability

We would like to emphasize again the scalability of TraSt. TraSt is not an unmodifiable application, but an open framework. It has completed the repeated work that must be done in the process of getting traffic statistics, such as open the device, grab packets, etc. It also provides the interface to obtain the detailed information of each packet for the test or evaluation process. In this way, testers can just focus on how to handle each packet, without having to care about how the packet be obtained. Therefore, TraSt is so powerful that it can be used to measure network performance (such as average TCP handshake time), monitoring the usage of the traffic by each application (according the port of each packet), etc.

4 Issues and Methodology

In this section, we clearly describe the issues to address and propose experimental methods.

4.1 Issues in 3G Charging

The 3G data charging system is a large and complex "black box". In this paper, we will get some detail information of the "black box" via external measurement. We explore four aspects:

1. Which layer in the protocol stack does each operator charging for?
2. Whether some protocol overhead is not taking into account?
3. What is the smallest units of charging for each operator?
4. Will smart phone receive packets passively without initiate a request? If so, will this part of traffic be charging?

The first two issues concern what to charge, the third concerns whether the charging result is accurate, and the last concerns whether there exist some irrational case in the charging process.

We need to describe in detail the last issue. In general, an application sends a request to the server, the server reply to it. This traffic should be charging of course. But what if the application doesn't send any request, and the server send packets to the device, will it receive the packets and will operators charge for this traffic? Obviously, it is unfair if the operators charging for this traffic. Even more frightening is that if the server is running a malicious program which continue to send packets to a user, will bring huge losses to the user.

4.2 Issues in Smart Phone Applications

There are many same types of applications on Android markets. The main problem is:

1. How great the difference is among the same kind applications?
2. What makes the difference?

These issues are what users and developers most concerned separately.

4.3 Methodology

We first study the issues in the operators' charging system. We conduct a series of experiments, get the data volume at the end device with TraSt, and compare it with that recorded by operators. We run tests with the three major mobile operators in China, which are China Mobile, China Unicom and China Telecom.

For the first issue, we open the network to run commonly used applications such as browsers and maps and get the data volume of each layer in the protocol stack (V_{UE}). Then close the network and get the operators' charging result (V_{Op}). Calculate the RE (Relative Error) of each layer:

$$RE = \frac{|V_{UE} - V_{Op}|}{V_{Op}} \times 100\% \tag{1}$$

We repeat the experiment three times, to see which layer's RE is the smallest. This layer's traffic is what the operators charging for. But at this moment, we don't know whether operators are charging for some special traffic, such as the traffic produce by TCP handshake. In order to make the ratio of special traffic as small as possible, we let the applications run enough time (10 minutes) so that they can receive/send enough data.

For the second and the third issue, we need to implement the smart phone application and the server ourselves. At this moment, we know which layer's traffic the operators are charging for, so we can deliberately let the client send specific protocol packets to the server, and observe whether the operators will charge for this traffic, and then we can get the result of the second issue. We can let the client send data as little as possible using the protocol that will be charged, and see how much the data volume recorded by the operators. Thus get the result of the third issue.

To make clear of the fourth issue, we just need to know whether the operators are using the NAT (Network Address Translator) technology, if so, what kind of NAT are they using? Only the Port Restricted Cone and Symmetric NAT have no such loophole [6]. If the address of the device is an internal IP, we know that the operator is using NAT. If so, we deploy two servers with different IPs running same service, let the smart phone application uses a same port to send packets to the two servers. Then the servers can obtain the source port of received packets. If the source ports are the same at the two servers, the operator is using Cone NAT, otherwise Symmetric NAT [6].

There are two methods to obtain data usage logged by operators. The first one is via the SMS from the operators, the second is to login the operators website and inquire data usage. China Mobile and China Telecom support both methods. But for getting finer result, we choose the second method for both of them. China Unicom SIM card used in our experiment only support the first method, the device will receive the billing SMS only when its traffic consumption since last billing SMS is greater than 50 KB.

After exploring the four aspect of the operators' charging system, we can further take evaluation of traffic consumption of smart phone applications. Because users and developers only concern traffic consumption charged by the operators, we could get more accurate comparison if we just taking this part of traffic into account. Table 1 lists the application kinds and corresponding applications to be evaluated. They are all content controllable applications. For the first issue, we take experiment in different scenarios on each type of applications, and use TraSt to get the part of traffic that is generated by the application and will be charged, and then we can compare the result among the applications. For the second issue, we can further analysis the traffic with the specific application, and try to get the root cause. Notice that we are evaluating content controllable applications. For the applications of the same type, we can make the data content basically the same at each network condition repeatedly.

Our mobile devices are HTC Desire and HTC Incredible S, running on Android 2.2 and 2.3.1 respectively.

5 Result

In this section, we will show our testing result for the operators' charging system and evaluation results for the four kinds of applications. All the following results are obtained in May 2012.

Table 1. Application types and applications

Browser	Build-in Webkit 533.1
	Opera Mobile 11.5.2
	UC Browser 7.9.3
	Firefox 10.0
Map	Google Map 6.3.0
	Baidu Map 2.1.0
	Sogou Map 2.1.0
Online Music Player	Xiaomi Music 1.4.3
	Duomi Music 4.0.1.00
	Douban FM 2.1.0
Video Call	Fring 3.9.3.33
	Tango 1.6.8607
	Skype 2.5.0.108

5.1 Charging System

Issue 1. Intuitively, all the three operators should charge for the same layer. We find that only China Mobile take Byte as statistical units. So we start with it. Figure 4 shows the $(1 - RE)$ of each layer by three experiments. It is obvious that the Network Layer's data volume is most close to the operator's charging result. The error is less than 1.5%. Therefore we can conclude that China Mobile is charging for the traffic of network layer. We can get the same result for China Unicom and China TeleCom, just by producing a large traffic (1MB is enough), so that the RE is different at each layer even the units of each element in formula 1 is KB or 10KB.

Issue 2. There are two main causees of the difference between TraSt and operators charging result. First, since experiment lasts long time, it is likely to occur packet loss. Second, there may exists some traffic that are not charged by the operators.

To verify the second cause, we use TraSt to count traffic consumption of each Network Layer protocol, including ARP, ICMP, ICMPv6 IPv4, IPv6 and others. Figure 5 shows the result. Since $217 + 160 = 377$, which is exactly the traffic consumption of IPv4. So we can conclude that China Mobile is not charging for ARP and ICMPv6. It is very hard to test for China Unicom and China Telecom because of the larger units in the bill. But we believe the results are the same.

Issue 3. For China Mobile we send a UDP packet with 1 Byte payload, its total length is 29 Byte. The charging result is 29 Byte. So the smallest charging units of China Mobile is Byte. For China Telecom, we send a UDP packet with 1 Byte payload two times at a same bearer. The charging result is 1 KB. So the China Telecom will add the packet length in Byte at each bearer and count the part less than 1 KB as 1KB. For China Unicom, it is more complicated since the

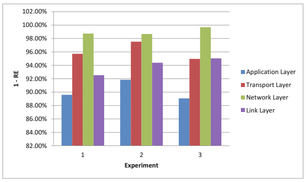

Fig. 4. 1 - RE of each layer

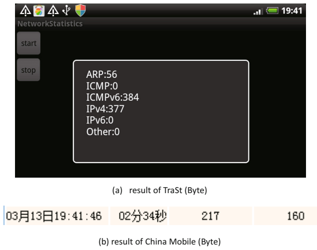

Fig. 5. Comparison of the result of TraSt and China Mobile

units on bill is 10 KB, but this does not indicate that China Unicom's charging units is 10 KB.

We let the client establish a TCP connection with the server at each bearer, and repeat it 20 times. The results of TraSt are (APR and ICMPv6 are removed): 184, 184, 360, 180, 273, 273, 180, 460, 630, 180, 180, 180, 180, 180, 180, 630, 327, 1770, 276, 313 (Byte). To make the total traffic consumption larger than 50 KB, we then run a Map application which consumed 255KB traffic. China Unicom's charging result is 0.27 MB. If the charging units is 1 KB, the result should be $1 \times 20 + 0.255 = 0.28$ MB. So the charging units of China Unicom is not KB. We guess it is also Byte like China Mobile.

Issue 4. The IP address of UEs are internal network IP (10.0.0.0–10.255.255.255 for China Mobile and China Telecom, 172.16.0.0–172.31.255.255 for China Unicom). So the operators are using NAT technology. We further take experiment mentioned in section IV.C and get the result that the operators are using Symmetric NAT. That means the smart phone won't receive any packet passively without initiate a request.

Table 2. Experiment scenarios for browser

Factors	Setting 1	Setting 2
Image	with image	without image
network condition	Wifi	3G
Buffer setting	turn off buffer	turn on buffer

Table 3. Experiment scenarios for map

Factors	Setting 1	Setting 2	Setting 3
Motion state	static	slow	fast
Buffer setting	turn off buffer	turn on buffer	

Table 4. Experiment scenarios for online music player

Factors	Setting 1	Setting 2
network condition	Wifi	3G
Signal strength	strong	weak

Table 5. Experiment scenarios for video call

Factors	Setting 1	Setting 2	Setting 3
network condition	Wifi to Wifi	Wifi to 3G	3G to 3G

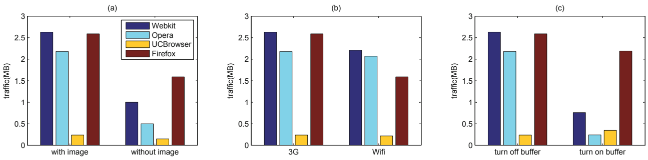

Fig. 6. Comparison of different browsers

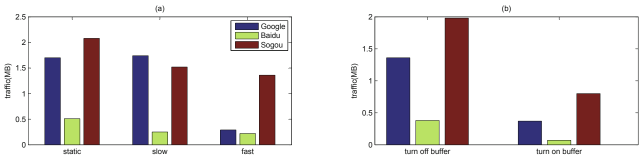

Fig. 7. Comparison of different maps

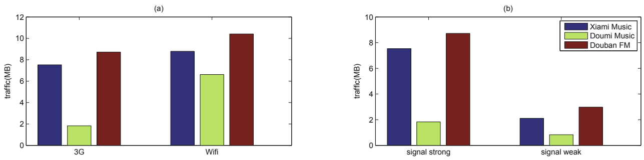

Fig. 8. Comparison of different online music players

5.2 Smart Phone Applications

We select specific scenario for each kind of applications to make comparative experiments. See in Table 2,3,4,5. For browsers, we let each application visit four website's homepage successively: http://www.youku.com, http://www.qq.com, http://3g.sina.com.cn/?from=www, http://map.baidu.com. For maps, in the static scenario, we search for "Poly Theater", find its location, set it as destination, and set "Peking University" as starting point. Then search for bus route. In the slow movement scenario, we open each map and then walk from the south-west gate of Peking University to the north-east gate. In the fast movement scenario, we open each map and travel from the south-east gate of Peking University to the west gate of Tsinghua University by bus. For online music play-

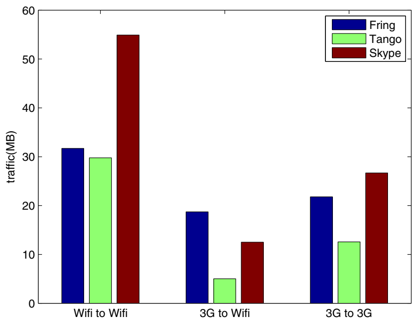

Fig. 9. Comparison of different video calls

ers we play music 10 minutes in each scenario for each application. For video calls, we use same application at tow phones to communicate 10 minutes.

Issue 1. The result of TraSt is shown in Figure 6 7 8 9. It is surprised to see that there exists such large difference among applications of same kind. For example, the UC browser's traffic consumption is nearly one-tenth of other browsers'! For the map application, Baidu Map's traffic consumption is relatively small, while Sogou Map's is relatively big. And Duomi Music is the most traffic saving among the three online music players. Tango consumed the smallest traffic than the other two video call applications in most scenarios.

Issue 2. If we make more in-depth analysis on traffic consumption, we could get the reason of why the gap is so big.

For browsers, because the UC browser didn't load any image when visited qq and youku. This could avoid a lot of unnecessary traffic consumption. Moreover, UC browser has done some optimizations in request mechanism to save uploading traffic.

For maps, different applications have different map source, many location details and information storage means are not the same. Google Map will only show the main building in the fast scenario. Sogou Map will update buffer automatically.

For online music players, Doumi Music will select smooth version in 3G network scenario to save traffic, and high-quality version in Wifi scenario. It also use buffer to save traffic. While the other two application doesn't support these features.

For video calls, although Tango is most traffic saving, the video quality is not good. It is saving traffic on the expense of user experience.

6 Conclusion and Future Work

The explosive growth of smart phones and rapid deployment of 3G networks have brought great convenience to people's lives. But there are still a lot of issues to be concerned. In this paper, we present TraSt, a framework that can record every detail information for each packet and help testers to do tests and evaluations on 3G networks and smart phone applications. We also do some experiments to explore four aspect of Chinese three major operators charging system and evaluate four kinds of popular smart phone applications. Our study offers some insight. For the operators' charging system, we could see that in general, the operators' charging results are quite accurate. The reasons why they are different from some traffic statistic applications' results are: first, some protocol overheads such as ARP are not charged by operators but counted by these applications; second, packet loss may lead some packets to be charged by operators but not received by the smart phone or vice versa; third, some services such as MMS (Multimedia Messaging Service), will produce traffic and be counted by these applications, but operators will charge it separately. For the applications of same kind, we could see their traffic consumption are always quite different. The reason is that the degree of optimization and user experience of different application are not the same.

At present, TraSt use file to store packets, when traffic is very large, it can not meet the real-time requirements. So in the future, we will use more efficient way such as interprocess communication to get packets.

References

1. Analysis Report on China's Smart Phone Market in March 2012, http://tech.hexun.com/2012-04-16/140429422.html
2. Android NDK, http://developer.android.com/sdk/ndk/overview.html
3. Jnetpcap, http://jnetpcap.com/
4. Ministry of Industry and Information Technology. 3G into the Large-scale Development Stage, http://www.miit.gov.cn
5. Tcpdump, http://tcpdump.org/
6. RFC 3489: STUN — simple traversal of user datagram protocol (UDP) through network address translators, NATs (2003)
7. 3GPP TS32.240: Telecommunication management; Charging management; Charging architecture and principles (2006)
8. G, A.: Global 3G Deployments UMTS HSPA HSPA+ (2010)
9. Huang, J., Xu, Q., Tiwana, B., Mao, Z.M., Zhang, M., Bahl, P.: Anatomizing application performance differences on smartphones. In: MobiSys 2010, pp. 165–178 (2010)
10. Koutsopoulou, M., Kaloxylos, A., Alonistioti, A., Merakos, L.F., Kawamura, K.: Charging, accounting and billing management schemes in mobile telecommunication networks and the internet, pp. 50–58 (2004)
11. Chan, M.C., Ramjee, R.: TCP/IP Performance over 3G Wireless Links with Rate and Delay Variation. In: MobiCom 2002, pp. 71–82 (2002)
12. Mahajan, R., Zhang, M., Poole, L., Pai, V.: Uncovering Performance Differences among Backbone ISPs with Netdiff. In: NSDI 2008, pp. 205–218 (2008)