Can We Pay for What We Get in 3G Data Access?

Chunyi Peng* Guan-Hua Tu* Chi-Yu Li Songwu Lu
University of California, Los Angeles, CA 90095, USA
{chunyip,ghtu,lichiyu, slu}@cs.ucla.edu

ABSTRACT

Data-plan subscribers are charged based on the used traffic volume in 3G/4G cellular networks. This usage-based charging system has been operational and received general success. In this work, we conduct experiments to critically assess both this usage-based accounting architecture and application-specific charging policies by operators. Our evaluation compares the network-recorded volume with the delivered traffic at the end device. We have found that, both generally work in common scenarios but may go wrong in the extreme cases: We are charged for what we never get, and we can get what we want for free. In one extreme case, we are charged for at least three hours and 450MB or more data despite receiving no single bit. In another extreme case, we are able to transfer 200MB or any amount we specify for free. The root causes lie in lack of both coordination between the charging system and the end device, and prudent policy enforcement by certain operators. We propose immediate fixes and discuss possible future directions.

Categories and Subject Descriptors

C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design—*Wireless Network*; C.4 [**Performance of Systems**]: *Design Studies*

General Terms

Measurement, Experimentation, Performance

Keywords

Cellular Networks, Mobile Data Services, Charging, Accounting

1. INTRODUCTION

Wireless access to data services is gaining increasing popularity in recent years, thanks to the rapid deployment of 3G/4G cellular networks. Statistics from OECD [25] shows that, 62% of broadband users in the US have subscribed to wireless data plans, with 137M subscribers by June 2010. On the front of mobile data applications, there are 1.2B mobile web users [23], and Facebook claimed more than 425M mobile monthly active users in December 2011 in its IPO statement. The explosive growth of smartphones (e.g., iPhone and Android phone) and tablets (e.g., iPad) will further accelerate this usage trend in the coming years.

While users enjoy wireless data access, it does not come for free. Most operators will charge the user a monthly bill based on the used data volume. The price for this usage-based charging ranges from 1s to 200s of cents for 1MB data in the US depending on the chosen data plans. Different from the flat charging scheme over the Internet, 3G/4G operators do not offer unlimited data usage for smartphone users. Both AT&T and Verizon effectively ended such data plans for new customers in 2011, and T-mobile limits the high-speed data volume in its so-called unlimited data plan.

The 3G charging system has been operational for a few years, and the practice has been generally successful. On one hand, the usage-based charging is not implemented without rationale. The radio spectrum is scarce and mostly licensed, and the offered access speed is bounded by the fundamental limits on channel capacity. On the other hand, the system mostly works as a black box for users, and users do have questions and concerns. Consider Alice, a typical 3G user, as an example. Alice just received a monthly bill of $25.8 for 387.4MB data usage, with a portion of the bill being shown in Figure 1. Even with this itemized data usage, Alice may still have lots of doubts and questions in her mind: (1) Does my iPhone really use 4385KB (but not 2.3MB less) since I remembered I only downloaded a 2MB app from the App Store? (2) How can I find out if the operator made a mistake and over-charged me? Anyway, I heard that up to 20M Americans using their iPhones/iPads are over-charged by 7-14% on average and up to 300% in some cases [26]. (3) For the 31KB item, I remembered I clicked an invalid web page link that did not show me any real content. Why should I be charged? (4) Is there any chance I can somewhat evade the charging system and pay less? The list goes on and long. On the technology side, answers to all these questions reside on the accounting[1] system used by 3G/4G networks.

Date	Time	To/From	Type	Direction	Msg/Kb/Min
03/04	09:21 PM	[redacted]	Internet/MEdia Net	Sent	31KB
03/04	08:07 PM	[redacted]	Internet/MEdia Net	Sent	4385KB
03/04	06:16 PM	[redacted]	Internet/MEdia Net	Sent	65KB
03/04	02:45 PM	[redacted]	MEdia Messaging	Sent	12KB

Figure 1: Example of itemized data usage.

In this paper, we present the arguably *first* work that assesses the 3G accounting system. Our evaluation criterion is *user centric*:

*The first two authors contribute equally to this work.

[1]We do not differentiate accounting from charging in this work by a slight abuse of wording definition.

MobiCom'12, August 22–26, 2012, Istanbul, Turkey.

We pay for what we get. We examine the usage gap between the operator-recorded data volume and the user-logged data amount. We conduct experiments with smartphones on two major US operational 3G networks, while also running similar tests in the third US carrier and two carriers in China and Taiwan. Using the phone-logged traces and the data volume recorded by the 3G accounting system, we analyze accounting behaviors in various scenarios. We focus on two aspects: (1) *How* is data access charged? It concerns the accounting architecture and its implementation within the 3GPP standards; and (2) *What* is charged? It attests the charging policy practice by operators.

Our study yields two main findings. First, we observe that *we be charged for what we never receive in certain scenarios*. The difference is generally small in the normal cases, resulting in about 10s to 100s of KB in typical applications. However, it can go up to 10s of MB for certain applications (e.g., video streaming). Moreover, the gap can grow quite large in extreme conditions. In one extreme case when a UDP session has no control loop, a mobile user receives data from this UDP session and roams into a no-signal zone. His ongoing UDP session continues to be charged by the 3G accounting system, despite no single bit ever received by the user. Our experiments show that this charging proceeds for more than three hours and results in 450MB or more in the charged volume! We also identify its root cause. It turns out that, current cellular accounting standards do not explicitly take feedback from end devices. Charging action is taken at the core components (e.g., GGSN/SGSN in 3G UMTS) inside the cellular infrastructure. The core components simply record the data volume traversing them to/from the given user for charging purpose. Consequently, whenever packet drops occur after traversing these components in the no-signal scenario, the charging system does not know the device status and overcharging may arise. The solution fix is to take feedback directly from the end device or access the device status information already collected within the infrastructure (e.g., at RNC) when making charging decisions. Second, we discover that *we can obtain what we want in data access free of charge*. The key is to exploit the free-of-charge service (e.g., DNS) by both operators and construct a "DNS tunnel" for other data transfer. Using a simple prototype, we are able to transmit 200MB or any amount we specify for free. The root cause is that, operators use application-specific charging policy, and loopholes exist in such policy practice. Operators do not enforce the full flow-based charging scheme by the 3GPP standard, but using only one or two fields in the five-tuple flow ID. Our work also shows that the policy enforcement is indeed operator specific. While all three US operators offer free DNS, the Chinese operator and the Taiwan carrier still charge it as usual. Though our experiments are mainly conducted in 3G networks, the observations are still applicable to 4G LTE networks since they follow almost identical accounting architecture.

The rest of the paper is organized as follows. Section 2 introduces the accounting and data charging process of 3G/4G networks. Section 3 describes the problem statement and study methodology. Sections 4 and 5 report cases where users are charged for data they never receive, and how to build toll-free data services, respectively. Section 6 describes other gray-area cases in charging. Section 7 further discusses architectural and policy issues. Section 8 compares with the related work, and Section 9 concludes the paper.

2. BACKGROUND ON DATA CHARGING

We first introduce 3G charging architecture for data services in context of UMTS, the most widely deployed 3G cellular network technology [17]. We then describe how charging actions proceed in the UMTS network core. Note that, the mechanisms and issues are also applicable in 4G Long Term Evolution (LTE) and 3G High-Speed Packet Access (HSPA) networks.

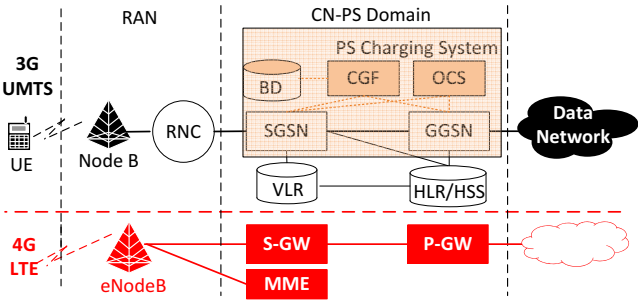


Figure 2: 3G/4G network architecture and charging components in PS domain.

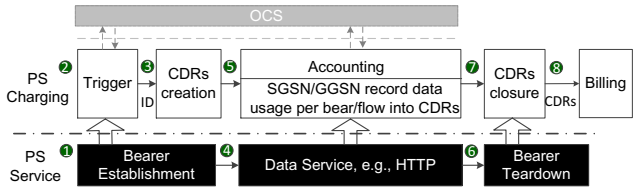


Figure 3: Charging process for a data service flow.

2.1 Data Charging Architecture

Figure 2 shows the overall 3G UMTS network architecture and charging system for data services. The UMTS network consists of the Terrestrial Radio Access Network (RAN) and the core network (CN). Its RAN includes the User Equipment (UE), the Node B, and the Radio Network Controller (RNC). RAN provides wireless access to UEs, and exchanges data session provisioning with the Packet-Switched (PS) core networks.

The major components of the PS core network are the *Serving GPRS Support Node* (SGSN) and *Gateway GPRS Support Node* (GGSN). SGSN is responsible for the delivery of data packets from and to the UEs within its geographical service area. GGSN serves as the hub between the SGSN and the external data networks, e.g., the wired Internet. It 'hides' the 3G UMTS infrastructure from the external network and acts as a router to a subnetwork. GGSN also performs charging, user authentication, and other functions.

In addition to SGSN and GGSN, three more charging components work in the PS charging system: the *Billing Domain* (BD), the *Charging Gateway Function* (CGF), and the *Online Charging System* (OCS). Current cellular networks support both offline and online charging modes [13]. In offline charging, data usage is collected during service provisioning in the form of *Charging Data Records* (CDRs), which are sent to the BD to generate data bills offline. SGSN and GGSN are responsible for collecting data usage and generating CDRs. CGF is used to validate CDRs from SGSNs/GGSNs and transfer CDRs to the BD. In online charging, mobile users have to pre-pay to obtain credits for data services in advance. The OCS authorizes whether or not users have enough credits. GGSN/SGSN deducts data usage from the available credits and stops data services upon zero credit.

2.2 Data Charging Process

We next describe how mobile users are charged for data services. Consider offline charging, and Alice is about to upload one photo

to her Facebook, thus starting a PS service (say, HTTP). Figure 3 illustrates the charging procedures during the data service process.

Initially, Alice has no available bearer service connection (which may carry one or multiple PS services). She thus establishes a bearer via Packet Data Protocol (PDP) Context[2] Activation [11] (Step 1). Upon this activation, the UE device is allowed to connect with the external data network through the SGSN and GGSN. This activation also triggers the charging procedure, and GGSN assigns a unique charging ID to the activated PDP context (Step 2). SGSN and GGSN then start to create CDRs using the charging ID (Step 3), and are ready to record the upcoming data volume. In addition to charging per PDP context, 3G also supports charging per data flow, called as *Flow Based Charging* (FBC). FBC separates charging for different services (e.g., web or VoIP) within the same PDP context [14]. One data flow is typically identified by the five-tuple: (1) source IP address or mask, (2) source port number, (3) destination address or mask, (4) destination port number, and (5) protocol ID of the protocol above IP, e.g., TCP or UDP [12]. For example, a HTTP data flow can be represented by *(*, *, *, 80, TCP)*[3].

Now Alice can upload her photo to Facebook. Both SGSN and GGSN route the UE's packets to/from the external data network during the data service session (Step 4). In the meantime, SGSN and GGSN record the traffic volume that arrives at them into corresponding CDRs (Step 5). Both SGSN and GGSN count the payload of GTP-U (GPRS Tunneling Protocol- User Plane) packets as data volume; GTP-U delivers data within cellular networks and runs below the IP protocol. Therefore, the data volume counts all packet headers above IP, including IP, TCP, and HTTP headers, but the MAC header is not counted.

The accounting procedure (Step 5) lasts until this data service completes. It occurs when the UE tears down this bearer (Step 6) in bearer-based charging, or when Alice closes her HTTP session in flow-based charging. CDRs are subsequently closed and transferred to the BD (Step 7). Finally, BD generates a billing item for the proper user based on the charging ID.

The online charging process is similar, though OCS participates in the triggering and accounting steps (Steps 2 and 5) by authenticating the GGSN/SGSN to use user credits. There is also no need to send CDRs to generate a bill since the consumed credits have been deducted (see Figure 3). In the paper, we focus on the offline charging, and the same issues also arise for the online charging.

2.3 On LTE

LTE is a 4G cellular network standard, offering even higher speed. Its architecture is similar to 3G UMTS. The major difference (also shown in Figure 2) is that the functionalities of RAN are performed by eNodeBs, whereas the functionalities of SGSN and GGSN are performed by *Mobility Management Entity* (MME), *Serving Gateway* (S-GW), and *Packet Data Network Gateway* (P-GW) [21]. Moreover, bearer establishment is supported by two procedures, i.e., *Evolved Packet System Bearer Activation* and *Public Data Network Connectivity Procedure* [16]. The charging system for LTE is almost identical to 3G UMTS, with S-GW and P-GW (replacing SGSN and GGSN) in charge of collecting data usage and generating CDRs.

3. PROBLEM AND METHODOLOGY

In this section, we identify the issues to address, and describe the experimental methodology.

[2]PDP contexts provide all the required information for IP packet data connections in cellular networks.

[3]Each of the five tuples can be a wildcard.

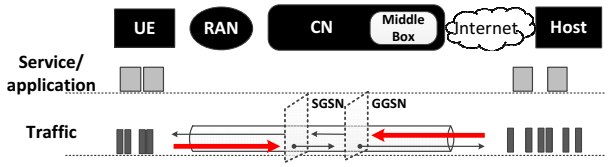


Figure 4: Current mobile data charging

3.1 Issues in 3G Accounting

In this paper, we examine the 3G accounting architecture, as well as the operators' policy practice. The research focuses on accounting, which records the usage volume over time for each user, rather than pricing that sets the unit price for usage and is driven by marketing and cost factors. We study the implication of such architectures and policies on user-perceived charges.

Our evaluation is to compare the user-recorded data volume with the network-recorded usage. Our study is user centric overall, in that *users pay for what they actually get at the end systems*. Anyway, the end systems are where users obtain their data service. This user-centric guideline may not necessarily concur with the infrastructure-oriented view taken by operators. When conflicts happen, one camp may suffer. In this work, our main goal is not to take specific position on what side to stand with, but to illustrate and quantify the discrepancy in different scenarios. To this end, we have built a simple tool, *BillAudit*, to conduct experiments between mobile phones and the Internet via the 3G network sitting in between. We collect the actual usage at the two end systems, and compare with the volume given by the 3G accounting system. The goal is to identify possible limitations and existing loopholes though such cases may occur rarely in reality, and demonstrate their effect on end users. We explore two dimensions of the problem:

- *How* to charge: How does 3G accounting handle various cases of end-to-end data delivery?
- *What* to charge: What is the difference in charging for different types of data traffic?

The first issue concerns the accounting architecture. It is about *how* to charge the mobile user in data services. As described in Section 2, the current 3G architecture takes the SGSN/GGSN based charging approach. It records how much data volume has traversed the intermediate SGSN/GGSN inside the 3G infrastructure (see Figures 2 and 3 for an illustration). This element-based charging takes the local view inside the 3G infrastructure, without coordinating with end devices when making accounting decisions. On the other hand, the data delivery path is always end to end. The end systems (e.g., the UE device or the server) may record usage volume different from what is logged inside the network. Note that the 3G accounting system does not explicitly collaborate with the end systems in its charging decisions. The effect is hence visible when failure or misbehavior occurs over the full delivery path.

In general, the end-to-end data delivery path consists of all six components (shown in Figure 4): the UE, the RAN, the 3G core network (CN), the middlebox, the wired Internet, and the host or server. Note that, the 3G network may deploy middle boxes (e.g., proxy servers, NAT boxes) over the delivery path, as shown in [30], and SGSN/GGSN resides at the CN. Assume that SGSN/GGSN and CN are always functioning. Any other component may fail. Specifically, we consider four cases in Sections 4 and 6: (1) The path segment between UE and RAN (i.e., the wireless delivery between Node B and UE) experiences problems in delivery; (2) The path segment between the Internet and CN has packet drops; (3) The path segment between the middlebox and the host breaks; and (4) The host or server is not accessible.

The second issue concerns on *what* to be charged. It depends on the charging policy. Each 3G operator can define its own application-specific policy on charging. Along this line, we are particularly interested in studying two cases:

1. Given certain type of free data services (e.g., DNS service discovered by our study) offered by operators, is it possible to exploit it to evade charges for other data services?
2. What is the current charging policy for application-level signaling or commands, which do not contribute to real content? Cases include FTP signaling over port 20, invalid HTTP links, HTTP redirects, and Email/IM signaling, etc.

Note that in the second instance, these signaling messages are not the actual content. Operators have every reason to charge it or not to charge it; it is not a right/wrong issue to address. In recent years, 3G/4G operators have been making effort to evolve from "dumb-bit-pipe owners" to "content/service providers". In the role-switching process, the charging policy may also evolve towards more content based; this is an interesting topic for future study.

3.2 Methodology

We conduct a series of experiments to examine the difference between the data volume recorded by operators and the one logged at the end device. In each experiment, we establish end-to-end data sessions from mobile phones to popular Internet services or our deployed server. We then record the data volume charged by operators and the ones observed at mobile phones or servers. We run main tests with two major mobile operators in the US, denoted as Operator-I and Operator-II for privacy concerns. They together offer nationwide coverage for 102.3M users, thus claiming about 50% of US market. For verification purpose, we also run similar tests with the third major US carrier that claims to support 4G LTE, a major carrier in China and Taiwan each. Our mobile devices use several Android phone models: HTC Desire, Samsung Galaxy S1/S2, and Samsung Stratosphere (that supports 4G LTE), running on Android 2.2, 2.3.4, 2.3.6 and 2.3.5, respectively. Our experiments show that all the findings are phone platform independent; this is not hard to understand. In our deployed servers, we use an Apache web server, a FTP server using Wing FTP software [10], and TCP/UDP servers written in Java.

We run experiments for both cases of extreme scenarios and normal settings. The extreme scenarios are carefully created in experiments, and seek to stretch out the charging system in worst-case settings. The normal settings capture users' common usage patterns, including popular protocols such as TCP, typical applications, and daily-life usage.

We use two methods to obtain data usage logged by operators. The first one is to dial a special number to retrieve the remaining monthly data usage in a near real-time mode. Most operators support this Dial-In feature, e.g., via dialing #DATA for Verizon, *DATA# for AT&T, and #932# for T-Mobile in the US. The data usage will then be delivered via a text message after this Dial-In. By logging data usage volume before and after our experiment, we obtain the usage volume observed by the operator during the experiment. The second method is to log onto the mobile operator website and access online data usage records (as shown in Figure 1). Operator-I only supports DIAL-IN method, while Operator-II supports both. We thus use the first method for Operator-I and the second method for Operator-II. In terms of report latency, we found that Operator-I may report data usage in five to ten minutes while Operator-II may take up to six hours to update their records. However, Operator-II provides an itemized data charging volume associated with the timestamp; a new item will be generated when a new PDP context (bearer) is established. We thus conduct experiments with proper time window (> 10 minutues) and establish a new PDP context for each experiment to avoid confusions and cope with latency. Both operators support 1 KB accuracy in data usage report. Since data usage logged by the operators only has timestamps, we need extra mechanisms to ensure that data usage belongs to the specific application or data flows in our experiments. For this purpose, we clean up the runtime environment (factory reset and disable "Background data" and "Auto-sync" functions). We also use monitoring tools (Wireshark [31]) to capture all-level packets to/from the phone. Whenever unwanted services observed, we re-run the experiment. For those tests that last for an extended period of time (e.g., three or six hours), we use Wireshark to filter out those unrelated packets in the trace analysis.

We use two tools to log data usage on mobile phones. The first is to use TrafficMonitor [7], a software tool available from Google Play. It records data volume for each application with 0.01KB accuracy. The second is to use our own tool, which is written via the TrafficStats class interfaces [8] in Android SDK to collect network traffic statistics. We record the number of packets and bytes transmitted and received on all interfaces and on a per-application basis. We use both tools to record the UE data usage and verify whether the usage is consistent or not. We further use Wireshark to log the traffic statistics at our server.

In each experiment, we record the data usage from operator V_{OP}, the one observed from mobile phone V_{UE}, and the one at server V_{SR} if used. We conduct each experiment for 5 –15 runs. The experimental results are quite stable in different runs. Due to space limit, we do not show results for individual runs, but only the average values unless explicitly stated.

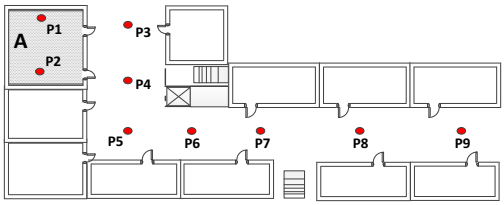
4. WE PAY FOR WHAT WE DO NOT GET

The first finding is that, we might be charged for data that never reach us or the data we never deliver to the destination. The root cause is that, the data volume is recorded inside the cellular network core without taking feedback from the end device when making accounting decisions; it can be different from the volume received at end device. In the rest of this section, we first describe the results in the extreme cases, which represent some worst-case behaviors and may rarely occur in reality. We then describe the average cases, which show how applications and users behave in common usage scenarios. We elaborate on how large the difference between the user's usage and the operator's charge can reach in these scenarios, explain their root causes, and suggest quick fixes.

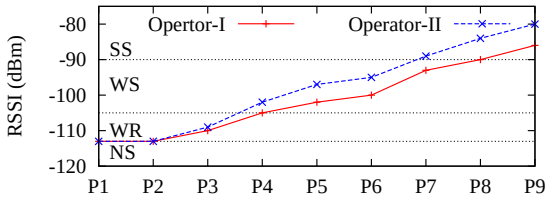
4.1 Extreme Cases

We first examine how bad overcharging can become in certain extreme conditions. The goal is to expose the potential downside of the largely successful 3G charging system. Note that these conditions do not represent the typical usage patterns in practice. They could occur in reality but only infrequently.

In these tests, data traffic is delivered using UDP between an Internet server and a mobile phone. Though most network applications run on TCP, recent traffic study [18] shows that, UDP is still used for data delivery in 10-20% applications, including video streaming, VoIP, and Virtual Private Network (VPN), etc. We consider downlink cases in both no-signal zone and weak-signal zone for the mobile device; these zones vary with locations. Our experiments show that, they are mainly caused by poor coverage by carriers. For example, we have experienced three to four dead zones or zones with very weak signals measured by RSSI on our office floor of the campus building. Mobile devices are unable to receive data when suddenly entering into a dead zone without signals. How-



(a) Experimental floorplan



(b) RSSI measurement

Figure 5: Our indoor testbed, where room A is a dead zone (NS-zone) for both operators.

ever, they are still charged though such data never reach them. In the worst-case scenario, we have observed that charging proceed for more than three hours and result in more than 450MB data if the application has no control loop!

4.1.1 UDP in No-Signal Scenario

We conduct DL-NS experiments to put our phone into a dead zone without signals, and see what happens to the ongoing downlink UDP transmission from an Internet server to the UE. The goal is to examine whether the data usage charged by operators differs from that received by mobile phones.

Our experiments are conducted in an indoor environment shown in Figure 5(a). The coverage varies at locations for both operators. Figure 5(b) plots the medium of the measured received signal strength indicators (RSSIs). RSSI values vary from -113 dBm to -80dBm[4] at various spots and fluctuate within 3 dBm at each spot. Based on RSSI values, we divide the whole area into four zones: (1) *SS-zone* with strong signals (RSSI $>$ -90 dBm); (2) *W-zone* with weak signals (-90 dBm $\geq$ RSSI $>$-105 dBm); (3) *WR-zone* with weaker signals (-105 dBm $\geq$ RSSI $>$ -113 dBm); and (4) *NS-zone* (i.e., dead zone) with no signals (RSSI $\leq$ -113 dBm). Note that, different operators yield different coverage strength; Operator-II has stronger signal strength than Operator-I in this setting. However, Room A remains a NS-zone for both operators. We also conducted prior experiments (e.g., making a phone call) to ensure that the phone is indeed out of service in Room A.

DL-NS Experiment Setting: Figure 6 illustrates how to set up the DL-NS experiment step by step. First, at P9 (i.e., in the SS-zone), we send a UDP request from the mobile phone to our own server to start this experiment; Once the communication is ongoing, the server responds with an acknowledge message to the phone and sets a timer, which triggers UDP data transmission upon timeout. Upon receiving the ACK, we move the phone from the SS-zone to the NS-zone (i.e., Room A) (Step 2), hopefully before timeout. Since it takes about 30 seconds to walk into Room A, the timer

[4]-113 dBm is the lowest signal strength that a typical mobile phone can receive; it implies that the phone is out of service. In indoor environment, strong signal strength is much smaller (here, (-90, -80) dBm) than the outdoor one that usually reaches -65 dBm. dBm stands for the measured power ratio in decibels (dB) referenced to one mW and 0 dBm equals 1 mW.

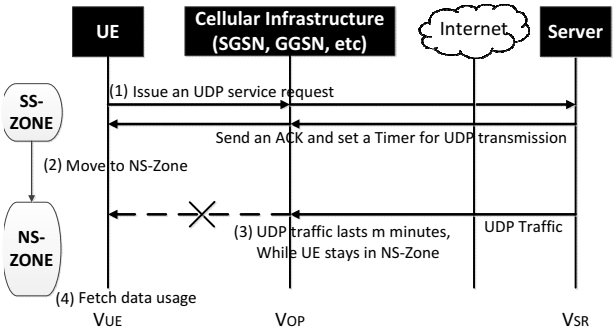


Figure 6: Procedure of DL-NS experiment.

is set as one minute to keep the server stay idle (no data delivery) during Step 2. Upon timeout, the server transmits UDP packets to the phone at a constant data speed s for another t minutes (Step 3); s and t are configurable parameters in the experiment. During Step 3, the phone remains in the NS-zone. We record data usage V_{SR}, V_{UE}, and V_{OP}, observed at the server, the UE, and the Operator, respectively.

Results: We first set the UDP source rate as $s = 50$ Kbps and the data transmission lasts for $t = 10$ minutes. Our server sends about 3.75 MB data ($50K \times 10 \times 60/8 = 3.75M$), similar to the volume charged by the operator (3.73 MB). The minor difference between these two volumes (i.e., $V_{SR} - V_{OP}$) is mainly caused by occasional packet loss. However, the mobile phone does not receive any such data, except the 80 B for one UDP request and one ACK message at the start. *This result shows that the charging infrastructure could charge mobile users of data that never reach them in case of a UDP-based application without control loop.* Moreover, we believe that, many mobile users might not be even aware of such a charge. It is quite common that mobile phone users unconsciously enter into a NS-zone in reality. They have no clue that roaming into the no-signal region may incur data volume charge by the operator, if the UDP sender is still transmitting.

4.1.2 Worst-Case Observations

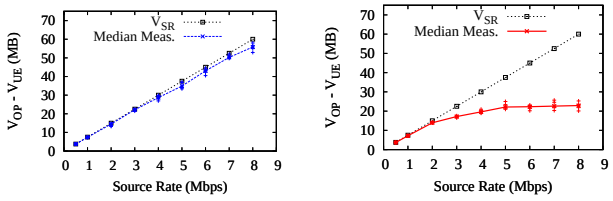
We test with various source rates and different durations. The gap between the operator charge and the volume received by users (i.e., $V_{OP} - V_{UE}$) can be approximated by $s \times t$, which is exactly the volume of data sent by the server but never reached the phone:

$$\boxed{\text{DL-Volume-Gap} = V_{OP} - V_{UE} \approx s \times t.}$$

Our experiments show that the approximation still holds even when the speed s goes up at least 8 Mbps or the duration t lasts three hours! The above finding shows that, the operators charge mobile users based on the data volume sent by the server and arriving at the cellular core network, but not the volume that cellular networks have actually delivered to the users. This rule still applies no matter how large the gap could turn into. For example, the operators have charged us for 450 MB in one run when the server keeps sending downlink data at 1 Mbps for one hour ($1 \times 60 \times 60/8 = 450M$), even though no single data bit arrives at the mobile phone!

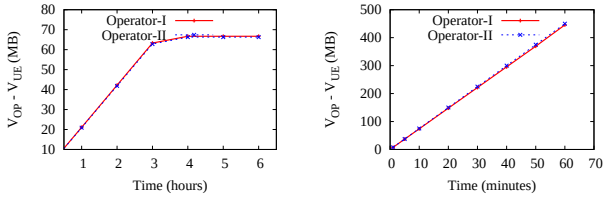
We change the source rate s from 50 Kbps to 8 Mbps to examine how the gap varies with high data speed. Figure 7 plots the DL-Volume-Gap for Operator-I using two sending servers with different link capacities. The transmission lasts one minute. The results are similar for Operator-II[5]. Note that in DL-NS experiments,

[5]The results for Operator-II will be omitted hereafter to save space if they are similar.



(a) Server 1 (b) Server 2

Figure 7: DL-Volume-Gap under various UDP source rates when $t = 1$ minute using two servers, where V_{SR} grows as $s \times t$, the medium values of DL-Volume-Gap, as well as each run observation (five runs), are given.



(a) $s = 50$ Kbps (b) $s = 1$ Mbps

Figure 8: DL-Volume-Gap with various in-NS-zone durations for 50 Kbps and 1 Mbps UDP flows.

the UE receives almost zero bits and the DL-Volume-Gap is approximately equal to the volume charged by operators V_{OP}. It is seen that the DL-Volume-Gap is in proportion to the UDP source rate s in Server-1 case (in Figure 7(a)). For Server-2, we find out that the gap is almost the same as V_{SR} when the data rate s is low ($\leq$ 2 Mbps); when the source rate increases ($>$ 2 Mbps), the one charged by operator is smaller. This is because Server-2 uses home Internet service and has bounded uplink speed. In contrast, using Server-1 with higher uplink bandwidth, the operator charges us for about 58.7 MB (close to $V_{SR} = 60$MB) in one minute at the 8 Mbps rate. *This test infers that, the operator charging practice is only based on how much data would arrive at the core network, no matter how fast it is.* Without much packet loss or congestion, DL-Volume-Gap grows in proportion to the UDP source rate.

Even worse, the operator may charge us for a long time. We put the phone in the NS-zone for different durations to see how long the gap may last. If the application layer tears down the session once the phone is out of service for a small duration of time, the gap would be small and not incur a large bill. However, if the application does not terminate, we find out that gap could last at least **three hours**! We run experiments for the slow session (50 Kbps) up to six hours and the fast session (1 Mbps) up to one hour. Figure 8 plots the DL-Volume-Gap when the in-NS-zone duration varies from one minute to six hours. Within the initial three hours, the gap for the slow session grows linearly with the duration t. Both operators stop recording when the usage reaches about 66.3 MB, which approximates about three-hour data transmissions. We do not run experiments for high-speed UDP sessions (e.g., 1 Mbps or even 8 Mbps) up to three hours, because the data usage probably goes up to 1.35 GB or 10.8 GB, which incurs a huge bill. In fact, the gap as large as 450 MB and the charging duration of about three hours are already significant enough.

Root Causes: We now explain the root cause. In the downlink case, traffic is delivered from the external server to the mobile device via cellular networks (e.g., GGSN, SGSN and RNC in turn). It is easy to see that, the observed data volume monotonically decreases along the downstream delivery path, i.e.,

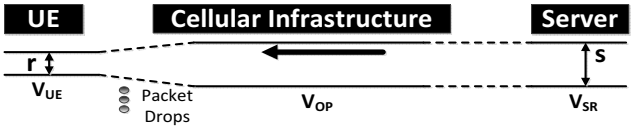
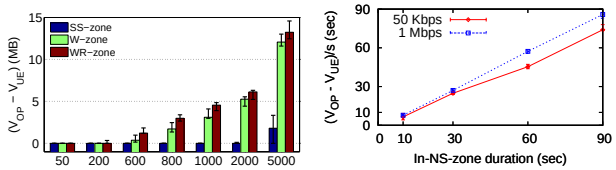


Figure 9: Illustration of DL-Volume-Gap creation in various wireless environments in DL-All experiments.



(a) Various signal strengths (b) Intermittent connectivity

Figure 10: DL-Volume-Gap in cases (a) under various signal strengths and (b) with intermittent connectivity.

$$V_{UE} \leq V_{SGSN} \leq V_{GGSN} \leq V_{SR}. \quad (1)$$

Due to unreliable packet delivery, packets might be dropped at any intermediate node, thus incurring the volume gap. In our DL-NS experiments, the last hop is broken, so no data would be delivered to the phone ($V_{UE} \approx 0$). As described in Section 2, the 3G/4G charging system obtains data usage based on the volume recorded by SGSN and GGSN. Therefore, those UDP packets, which arrive at GGSN or SGSN but never reach the UE, are still counted as the data usage by this UE. This results in a large gap between the actual data usage and the billing volume.

4.1.3 *Still-Bad Case: Even With Signals*

We next show that, the charging gap still exists even when the wireless link is not broken. The gap concerns the wireless environment in terms of available radio link rate. We conduct another DL-ALL experiment, where the mobile phone is statically placed in different zones with various signal strengths. Different from the DL-NS experiments, UDP packets are immediately transmitted once the handshake between the phone and the server is established.

Figure 10(a) plots the DL-Volume-Gap when the phone is placed in zones with different signal strengths under various source rates in Operator-I. Each data transmission lasts one minute. The figure shows that, *the DL-Volume-Gap becomes larger as the signal strength becomes weaker or as the source rate becomes larger*. Figure 9 illustrates why it happens, and Table 1 shows the detailed results for three examples of experimental traces.

We make three observations. First, the core network receives almost all data packets (i.e., $V_{OP} \approx V_{SR}$). The charging gap is still caused by the unsuccessful packet delivery from RAN to UE. Second, packets are dropped in RAN because the incoming source rate is much higher than the effective rate r of the wireless link to the phone (see Figure 9). The effective rate depends on wireless signal strength. For example, the effective rate is 168.1 Kbps in the WR-zone, much smaller than in the SS-zone (about 644.4 Kbps). Third, not all the mismatches between the source rate s and the effective rate r lead to packet drops. Take the example of SS-zone with $s = 800$ Kbps. It spends more time (about 74.7 seconds) and incurs large delay. We infer that this attributes to the buffer mechanism, which temporarily stores incoming packets (if too fast) and retransmits them if needed. However, as the source rate further increases, the speed mismatch becomes too large (especially, in the

Setting	V_{UE} (MB)	V_{OP} (MB)	V_{SR} (MB)	Link rate r (Kbps)	Finish time (sec)	DL-Volume-Gap (MB)
SS-zone (-84 dBm)	6.0	6.0	6.0	644.4	74.7	0
W-zone (-98 dBm)	2.90	6.0	6.0	326.7	71.0	3.10
WR-zone(-109 dBm)	1.46	6.0	6.0	168.1	69.5	4.54

Table 1: Example results for three DL-All experiments when source rate is s =800 Kbps and $t = 1$ minute.

WR-zone/W-zone) to be handled by buffers, leading to eventual packet drops. *Consequently, we still pay for bits that never reach us even though wireless links exist. The charging gap depends on the operating environment.*

We also consider the case with intermittent signals where mobile users may lose signals for a while but recover them shortly. This scenario is common with cases of mobility and special landscape (mountains or high buildings). Our findings show that, those packets that the phone miss in NS-zones still contribute to the charging gap, though the communication recovers soon and buffering and retransmission mechanisms reduce the charging gap to some extent. Figure plots the DL-Volume-Gap when the phone loses signals for 10, 30, 60, 90 seconds. In the meantime, the UDP server sends packets at a speed s. We can approximate the data volume that arrived in t time but finally received by the phone is $V_{back} = s \times t - (V_{OP} - V_{UE})$. Figure 10(b) plots $(V_{OP} - V_{UE})/s$ (i.e., $t - V_{back}/s$) under various in-NS-zone durations. The larger the duration, the fewer the received packets. The results imply that, buffering and retransmission do contribute to packet delivery (retrieving 15 out of 90 seconds data in 50 Kbps-UDP session). However, those packets not recovered are still counted into the mobile bills. *It also shows that the gap exists even when mobile users only lose signals for several seconds.*

4.2 Common Cases

We now study the common cases, which reflect the usage patterns by applications and users in their daily activities. Our study has three categories. The first is to see how TCP, the dominant transport protocol for applications, reacts in the no-signal and weak-signal zones. In the second category, we study five popular applications, including Web browsing, Skype for VoIP, YouTube, PPS streaming, and streaming over VPN tunnels. In the third category, we report the user-based, weekly accounting gap.

4.2.1 TCP Cases

We next study the charging behaviors of TCP flows, which turn out to be not too bad in terms of the overcharged volume. Intuitively, TCP behaves differently due to its built-in mechanisms of congestion control and reliable data transfer. Its feedback loop offers implicit coordination between the network and end devices. We conduct DL-NS experiments via TCP, but let the server deliver packets without timeout. We keep the mobile phone longer in the SS-zone before roaming into the NS-zone. As expected, the DL-Volume-Gap greatly reduces; it is seen to vary between 2.9 KB and 50 KB in our experiments. As we know, the charging gap is determined by how many bytes are delivered before automatic TCP session teardown. This is determined by the congestion window size before the UE enters into the NS-zone and the timer for automatic teardown. When the phone is out of service, packets in the congestion control window are still sent out; Since no more packets can be acknowledged and the window decreases; the unacknowledged packets are retransmitted until automatic connection teardown. Note that, some ACKs at the UE fail to be sent out due to the broken connection. Figure 11 plots an example of TCP segment records observed by the two ends (the gap is 45.7 KB here).

In reality, most current Internet applications are built on the top

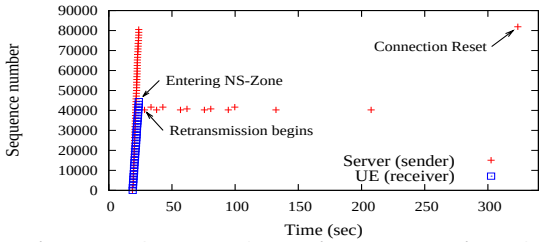


Figure 11: One DL-NS experiment trace using TCP.

of TCP. The inherent congestion control and automatic connection reset may release this connection once it fails for an extended period of time. This is why the charging gap is not large in reality. However, this is not the best practice from the 3G accounting perspective. It has to rely on higher-layer protocols such as TCP to handle abnormal behaviors of both connection teardown and overcharging. Moreover, UDP-based multimedia or other applications may still suffer if they do not implement control logic for automatic session teardown.

4.2.2 Application Behaviors

We carry out DL-NS experiments using five applications, including Web browsing, Skype, YouTube, PPS streaming [5], and VLC streaming [9] over VPN tunnels. In the web browsing test, we visit `www.cnn.com` at different times of the day; In Skype test, we make Skype video call, which uses UDP-based data delivery with built-in rate control [32]. Other three applications provide video streaming on phones. YouTube is TCP based, while PPS and VLC streaming are UDP based. PPS streaming is a very popular peer-to-peer application in China, whereas VLC + VPN offers one method to watch home HD TV. These applications have built-in control mechanisms, which can reduce rate or even tear down data delivery if the network connection degrades or breaks. We run each application for 5-15 runs with two US operators, except that VLC+VPN is blocked by the firewall of Operator-II. We start these applications in a SS-zone and enter into a NS-zone in several minutes.

Table 2 records the observed volume gap ($V_{OP} - V_{UE}$). The gap is negligible for Web browsing, and no more than 1MB for each Skype or Youtube run. However, it may reach up to 4.3 MB and 29.9 MB for PPS and VLC+VPN streaming, respectively. Interestingly, we observe that the gap be negative (but close to zero) for Web browsing; we figure out that it is because some packets (i.e., DNS packets) are free of charge, to be elaborated in Section 5. The difference between Web volume-gap of two operators is caused by different versions of Web pages. Mobile CNN page (about 0.2 MB) is fetched for Operator-II while the official CNN page (about 1.2MB) is fetched for Operator-I; the percentage of the volume gap is about 1.5–2.5%.

We also find that the volume gap varies significantly even for the same application (except Web browsing). For example, the gap varies from 0.1MB to 0.99MB for different Skype runs in Operator-II. It turns out that the average transmission rate during the ten seconds before entering into a NS-zone varies from 221 Kbps to 1.0 Mbps. We observe that the gap be bigger if the transmission rate before going into the NS-zone is larger. It is easy to understand, since the volume gap is contributed by the source rate s and durations t, where s is the average source rate during the period in the NS-signal zone and determined by rate control for each application. We also observe large gap for VLC+VPN. This is because its automatic teardown timer is larger. The teardown timer in VLC+VPN lasts from 30 seconds to several minutes (about 6-minute value was observed in our experiments, leading to 29.9 MB charging gap),

	Operator-I			Operator-II		
APPS	Med (MB)	Max (MB)	Min (MB)	Med (MB)	Max (MB)	Min (MB)
Web	-0.03	-0.04	0.00	-3KB	-4.6KB	0.6KB
Skype	0.88	0.99	0.40	0.68	0.99	0.10
YouTube	0.23	0.34	0.20	0.44	0.63	0.36
PPS	3.30	4.3	0.72	1.4	1.6	0.92
VLC + VPN	2.97	29.9	1.45	-	-	-

Table 2: Volume gaps for applications in DL-NS experiments.

whereas it is merely several seconds in Skype. These application tests again demonstrate that, though mobile data charging is largely successful in practice, nonnegligible overcharging is still observed due to problems in the current architecture.

We conduct another experiment to assess the performance over intermittent wireless channels. We watch videos via VLC streaming when we roam around the office area with several NS-zones. The wireless signals are intermittent, but usually recover within minutes. The video halts when we lose signals, but resumes once the wireless link is reestablished. In our experiment, we see that this video streaming never tear down and the observed volume gap reach up to 27.7 MB. Moreover, the gap depends on the number of SS-NS zone switches during our movement. We observe 11.8 MB and 27.7 MB charging gap for 10-minute and 30-minute movements, respectively. In DL-NS experiments, the observed gap for VLC streaming is at most 2.97 MB since the server tears down video streaming upon losing responses from the mobile client.

4.2.3 User-Based Usage Scenarios

We also study the accounting discrepancy for seven users (university students), who have data plans with two US carriers. We record the data usage observed by the mobile phone and the one charged by the operator for two weeks (June 10 - 23, 2012), except that User 7 had only one-day record on June 22. Note that, this small-sample user study may not well represent the common cases of daily usage for average users in our society. Instead, we intend to demonstrate how much the charging gap could be observed in reality, which depends on executed applications, usage patterns and locations. Among these users, the most popular applications are Web browsing and Gmail. Table 3 also shows other popular applications for each user, such as Gmap, Skype, YouTube, PPS, FaceBook, ebook reading, and games. Though data usage varies with users (from 47.1 MB to 900.2 MB) due to user behavior diversity, the volume gap is indeed small ($<$ 1 MB usually) for most users in reality. Big volume gap is not commonly observed in practice due to the built-in control mechanisms in many applications and infrequent encounters of NS-zones. However, we still observe that Users 4 and 7 have experienced volume gaps as large as 5.3% and 7.2%, respectively. User 4 once watched VLC streaming three times during a day while staying and roaming around his office area with several NS-zones; User 7 watched video using YouTube or PPS on the train to/from New York City, where there is a long tunnel without signals. During the round trip, User 7 transmitted and received 72.4 MB, but was charged by the operator for 77.6 MB, with the gap being 7.2%.

4.3 Recommended Quick Fix

We now recommend quick fix to the overcharging issues. The fundamental problem is that, the 3G network takes an SGSN/GGSN-based charging approach, which only records the data volume traversing these intermediate steps on the end-to-end delivery path. They do not coordinate with end devices when making accounting decisions. Specifically, they never take explicit feedback from the end systems. Therefore, when failures occur over the downstream path after SGSN/GGSN, the charging system

	Operator-I				Operator-II		
User	1	2	3	4	5	6	7
Apps (excl. Web, Gmail)	Gmap	Stock Games	Skype, PPS FaceBook	YouTube PPS	ebook	-	YouTube PPS
V_{UE} (MB)	194.2	270.3	124.6	900.2	121.7	47.1	72.4
V_{OP} (MB)	192.6	270.0	129.4	948.4	120.9	47.3	77.6
Gap (MB)	-1.8	-0.3	4.8	48.2	-0.8	0.2	5.2
	-0.9%	-0.1%	3.9%	5.3%	-0.6%	0.4%	7.2%

Table 3: Volume gap for user studies during June 10-23, 2012 (User 7 had only one-day usage record on June 22, 2012).

is not aware of the status of the mobile device, thus incurring overcharge. We now suggest three feedback mechanisms that help to remedy the problems. Note that our proposals are also applicable to packet drops due to weak signals, not only in NS-zones.

In the first proposal, the charging system takes explicit feedback regarding the status of the end device. For the downlink case, our solution can be implemented within the 3G infrastructure without interacting with the UE. We use the feedback from RNC to obtain more accurate data usage delivered to the UE device. We use the field "*RNC Unsent Data Volume*," which records the data volume not delivered to UE, defined by the 3G standard [15]. RNC reports this record to SGSN, which computes the data volume successfully delivered to UE, i.e., $V_{succ} = V_{SGSN} - V_{RNC\_unsent}$. It thus enables the operator to charge the user based on the data volume delivered to UE. This way, the huge DL-Volume-Gap (e.g., the 450 MB) can be eliminated.

We next fix problems with the session teardown in the absence of signals, where the charging can stop early to avoid overcharging. Our suggested solution is to deactivate the PDP context soon after the UE device cannot be reached. This can be implemented by the soft-state mechanism on the PDP context. We set a timer with the PDP context for UE, and the timer, as well as the PDP context, will be refreshed via the data delivery to/from UE. Note in three-hour DL-NS experiments, PDP context is not released in time because there is incoming traffic associated with it. This implies that the operator probably makes wrong decision that the PDP context should be kept alive. We thus suggest refreshing the timer based on actual data delivery, or the paging of UE when the actual data usage is zero. This offers the 3G charging system an alternative feedback mechanism on the UE status, but may incur excessive control overhead.

The third feedback mechanism also helps to reduce overcharging. Whenever big data usage is generated, it should trigger an exception verification to check whether the charging makes sense or not. For example, when a data session lasts for an hour or produces about 100 MB data, the 3G core network should verify whether it is indeed normal charging practice. This can be done by sending a signaling message to RNC to query whether the UE status is normal. The RNC subsequently reports the UE status and facilitates SGSN/GGSN in its charging decision.

We note that Cisco has proposed overcharging protection for GGSN to be aware of lost radio coverage using the feedback of SGSN [1]. It is to assess the device status due to lost coverage.

4.4 For Other Carriers

We also run similar experiments with three other major carriers, one each in the US, China and Taiwan. All the observed results still hold in general. The minor difference is that, (1) three-hours charge for a 50 Kbps UDP flow in DL-NS experiments is observed for two major US operators, while at least 5.7 hour charge is observed for the third US operator, one hour charge is observed in China, and about 42 minutes occur in Taiwan; and (2) the maximal UDP source rate is smaller in China, e.g., the transition point in Figure 7(b) happens at 1 Mbps for the Chinese carrier. This is

because the data rate supported over the wired Internet is smaller in China. We also conduct two-week usage studies for one user in China and two users in Taiwan; their observed gaps are negligible (<1 MB, within 0.5% error) because they mainly use them for Web, Gmail and SMS exchanges and the overall volume is small.

5. WE GET WHAT WE WANT FOR FREE

The second finding is just the opposite. We can take free rides to obtain "toll-free" data services without incurring any charge. The root cause is that, the current charging policy practiced by operators has loopholes, and can be exploited to build "free" data services. Our study shows that, both operators offer free Domain Name System (DNS) service via transport-layer port number 53. There is almost no enforcement mechanism to ensure that the packets going through this port are indeed DNS messages. Even worse, no effective mechanism exists to limit the traffic volume going through this port. Consequently, this free service can be readily abused to create "toll-free" data services. We have built a simple prototype, and demonstrated that it is feasible to offer various data services, e.g., file downloading or video streaming, through a special proxy server relaying data over the free transport-layer port. The process is similar to calling 800-like voice hotlines, but for free data access.

5.1 Loopholes in Charging Policy Practice

The 3G standards offer carriers enough freedom to define their own charging policies. Our experiments show that DNS packets are not charged by operators. We sent out 100 DNS queries (about 18KB in volume), and operators did not charge the incurred data usage. The operators' practice is also easy to understand, since DNS is a fundamental service to jump start Internet applications. Operators thus have every reason to offer it for free, to facilitate followup data usage by the services. Thus, free DNS service is well justified as a good policy practice.

However, the operator practice to offer free DNS service does have loopholes. Internet RFC 5966 stipulates that DNS service is offered using transport-layer port 53 via UDP or TCP [27]. To identify a data flow, the 3GPP standards define five-tuple flow ID composed of source and destination IP addresses, source and destination port numbers, and protocol ID (see Section 2). However, both operators do not strictly enforce this service via the standard five-tuple flow ID, but via only the destination port (plus protocol ID for Operator-II), thus exposing a loophole.

DNS-TEST Experiment Setting: We test DNS-related charging in five cases: (1) *DNS-Default*: send 100 DNS queries to the default DNS server provided by the operators; (2) *DNS-Google*: send 100 DNS queries to a Google DNS server (IP address: 8.8.8.8); (3) *TCP53-Google*: repeat (2) but via TCP at port 53; (4) *TCP53-Server*: send 50 random packets to our own server using TCP via port 53, and request the server to return the received packets; each packet (including IP/TCP headers) is 1KB; Source port number is randomly chosen; and (5) *UDP53-Server*: repeat (4) but using UDP. The goal of these experiments is to verify what factors the free DNS service depends on: (1) Does it depend on the server address? For example, is DNS free only via the operator DNS servers? (2) Does it depend on the protocol ID? For example, is it free for both UDP and TCP? (3) Does it depend on the source port number or check the DNS message semantics?

Results: Figure 12 plots the data volume observed by UE and the amount charged by Operator-I and Operator-II in all five cases. The results shows that,

> Operator-I: Packets via **port 53** are FREE
> Operator-II: Packets via **UDP + port 53** are FREE

Specifically, UE sends and receives about 18.1 KB for 100 DNS queries and responses in both *DNS-Default* and *DNS-Google* tests. In the *TCP53-Google* test, the traffic volume increases to 48.1 KB due to TCP signaling overhead. In both *TCP53-Server* and *UDP53-Server* tests, the UE works as expected when sending/receiving 100 KB. Operator-I charges for free (i.e., $V_{OP} = 0$) in all cases while Operator-II charges those TCP cases. From these results, we learn that the free DNS service is implemented by Operator-I using only one field in the flow ID (i.e., the destination port 53). In contrast, Operator-II uses two tuples in the flow ID, i.e., UDP over destination port 53.

5.2 Building "Toll-Free" Data Services

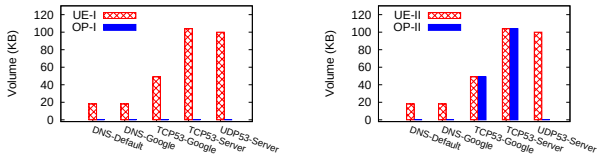
We now exploit the loopholes in the free DNS service to enable "toll-free" data services. We deploy a proxy server (placed outside the cellular network), which exchanges data services with mobile phones through the 3G carrier. We use "DNS tunneling" between the phone and the proxy. Data communication between the proxy and mobile phones is carried in UDP at port 53 by encapsulating data packets in DNS messages, which traverse the 3G network free of charge. We can also build the DNS tunnel using TCP at port 53 for Operator-I. The design can be readily extended to communication between UE and an Internet server, where the proxy server will act as a hub to relay packets on behalf of both UE and the server. The idea of DNS tunneling is also used in the iodine tool [4], which is designed for data access in different scenarios where DNS queries are allowed but the Internet access is blocked.

We run experiments using the prototype to demonstrate that free data service is feasible. Figure 13 plots V_{UE} and V_{OP} for three scenarios: (I) UE sends one request to download a 5MB video clip from a public website, (II) UE uploads a 3MB file to our server, and requests to return the delivered packets, and (III) UE sends many small requests (100 B) to our server for an hour, and each demands a 1KB response. These cases are to validate whether our service supports unbounded traffic upon a single request, whether it supports large-volume uplink and downlink traffic, and whether it allows for long-lived sessions.

Our results show that, both operators can be exploited for free data services in these scenarios, except that Operator-I does not allow unbounded traffic for a fake "DNS" request. In the first test, Operator-I only allows to deliver 29 KB downlink data to the UE, while Operator-II delivers much larger file (up to 4 MB). We gauge that Operator-I might have enforced checking to verify the size of the response message, in which a DNS message size is typically bounded. However, this checking can be easily bypassed. The UE simply sends out many small, dumb packets over this session to increase the quota for downlink traffic. Then large downlink data can pass this checking. This has been validated in scenarios (II) and (III). In these tests, the gap between V_{UE} and the expected file size is mainly caused by unreliable transmissions via UDP. We can implement reliable transfer mechanisms over UDP to eliminate this loss. Since each scenario has ten or more experimental runs at the data rate from 100 Kbps to 1 Mbps, the total free data we have obtained from this DNS hack exceeds 200MB[6].

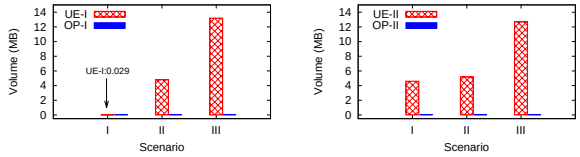
In addition to DNS hacking, other tricks exist for free data services by exploiting the loopholes in the charging policy. For example, in case some operators offer free Internet access to a given website, Web redirection from one free Web server to the target Web page is used to enable free data services [3]; using certain, free Access Point Name (APN), which is a configurable network

[6]We do not run into legal issues since we had unlimited data plan from Operator-II in the past and bought unlimited daily plan from Operator-I in our experiments.



(a) Operator-I (b) Operator-II

Figure 12: V_{UE} and V_{OP} in DNS-TEST experiments.



(a) Operator-I (b) Operator-II

Figure 13: Feasibility test of free data services.

ID used by a mobile device when connecting to a carrier, offers another way for free data service in some carriers, e.g., AirTel India [2] and UK Three [6]. These examples, including our DNS hacking, show that policy offers flexibility but may also be abused if not enforced properly.

5.3 Recommended Quick Fix

The simplest fix is to stop offering free DNS service or other forms of free data services inside cellular networks. Beyond this option, we suggest three possible remedies to this loophole while retaining free DNS service. The first solution is to enforce checking on the IP address of the DNS server. Therefore, free DNS services are only allowed if these messages go to designated DNS servers provided by operators or other DNS servers authorized by the operators. This enforcement eliminates the possibility to go through those fake DNS servers. The possible downsides include: extra effort is needed to authenticate DNS servers, not all DNS servers across the Internet can be directly accessed by UE, and workload at the designated DNS servers increases. Unfortunately, it is still possible for malicious users to deceive those resolvers/servers to forward fake DNS requests to a fake DNS server, but the cost is higher. The second fix is to provide quota for free DNS service. The quota for free DNS data can be set on a per-UE, per-week (or daily) basis in advance. The data usage beyond DNS quota will still be charged. The challenge for this approach is how to set an appropriate quota. Ideally, the quota should be estimated based on both normal usage and sudden surges. The third approach is to enforce message integrity check to verify the authenticity of each DNS message. However, it incurs excessive processing overhead on a per message basis.

5.4 Carriers in Other Regions

We run similar tests with other carriers. We indeed observe that the free DNS policy be operator dependent. The third US operator also offers UDP-based DNS for free, and behaves similar to Operator-II. However, for both carriers in China and Taiwan, the DNS service is not free. Both operators charge DNS messages identical to data traffic.

6. GRAY AREAS IN DATA CHARGING

We now describe charging cases in gray areas, where the users may be charged differently by the operators, compared with the actual data usage perceived at end hosts. However, there is no simple, accepted charging rule in these cases. We will show that the users may be charged for wrong or careless uplink operations; we also examine the impact of the middlebox deployment and Internet traffic congestion on mobile data charging. We finally assess charging on application overhead.

6.1 UDP Uplink to a Nonexistent Host

The worst uplink case is to use UDP packets to a nonexistent host (i.e., no packets can be successfully delivered). Our tests show that both operators still charge every bit sent by UE. The root cause still lies in the SGSN/GGSN based charging architecture, similar to the downlink case of Section 4. The good news is that this scenario is not very common unless the device is hijacked.

We also test UDP uplink traffic to our server under various wireless environments. It turns out that, there is no (obvious) gap between the data volume arriving at the receiver (i.e., our server) and the volume charged by the operator. However, our UE traces show that the UE does retransmit data over the wireless link (particularly in the WR-zone/WS-zone), i.e., $V_{UE} > V_{OP}$. Note that, these retransmitted data over the wireless link will not be observed by SGSN/GGSN, thus incurring no extra charge beyond those volume perceived by the receiver. The same conclusion also holds for uplink TCP sessions.

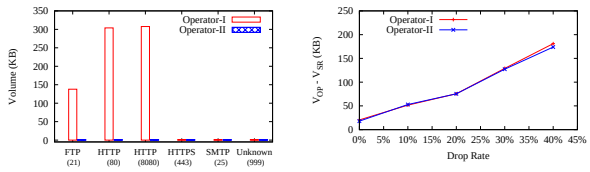
6.2 Effect of Middle-boxes

Middleboxes (e.g., proxy servers, CDN servers, NAT boxes, and firewalls) can be deployed inside 3G/4G networks for performance enhancement or extra service [30]. Indeed, our study confirms that proxy servers are placed in 3G networks. We find out that, Operator-I deploys proxy servers to handle popular applications, including HTTP and FTP. Consequently, the end-to-end data session between the UE and the server is split into two segments, one between the UE and the proxy, the other between the proxy and the HTTP/FTP server. We now assess the impact of such session splits due to middleboxes on charging. In the worst case, the UE interacts with the proxy rather than its intended server, and incurs overcharging. The user is charged though (s)he never receives any service! This can be illustrated by the following experiment.

We let the mobile phone connect to a non-existent host (i.e., an unallocated IP address) and upload a 1 MB file using TCP. Since the host does not exist, it should stop early (e.g., after several TCP SYN requests). To our surprise, we discover that, the data sessions for HTTP (80, 8080) and FTP (21) last much longer than expected. The delivered data volume for HTTP and FTP reaches 300 KB and 130 KB, respectively, in Operator-I. We also test with other popular applications, e.g., HTTPS(443) and SMTP(25), and unknown ports. Figure 14(a) plots V_{OP} (the same as V_{UE}) using different port numbers for both carriers. In Operator-I, we examined the TCP traces collected at the UE, and observed that those sent packets be acknowledged by TCP, though the IP address on the server side is nonexistent. This indicates that at least a middlebox has been deployed on the delivery path, which responds to UE requests on behalf of the server. The bad news is that, Operator-I also imposes charges though such HTTP/FTP data never reach the server side! We note that, the proxy is operator dependent and application specific. Other popular applications do not observe such proxy servers. Operator-II does not seem to have deployed such middleboxes in its core network even for HTTP/FTP.

6.3 Packet Drop over the Internet

Charging discrepancy exists when packets are dropped over the wired Internet, which is outside the cellular core network. We illustrate the scenario in Figure 14(b). The UE sends 200 KB data to



(a) Middlebox (b) Packet drop

Figure 14: Results when (a) connecting to nonexistent hosts using different TCP ports and (b) experiencing unreliable Internet packet delivery.

the Internet server via the cellular network, but packets are dropped right before reaching the server. Note that the Internet only offers best-effort service, so IP packet drops can be common, e.g., upon network congestion or malfunctioning routers. In this case, packets have been through the cellular network and recorded by SGSN/GGSN for charging. In our experiment, we upload an image file to our server from the mobile phone. We vary the packet-drop rate from 0% to 40% before reaching the TCP receiving end; we use it to emulate packet loss on the Internet. Since TCP will retransmit these lost packets from the UE, they incur additional charging volume at the carrier. We plot the charging gap (between the server and the operator) versus the drop percentage in Figure 14(b). The figure shows that, the charging volume increases almost in proportion to the drop rate. For operators, this makes perfect sense since the 3G network does deliver those packets. However, end users never receive those dropped data. The same phenomenon occurs for uplink UDP transmissions.

6.4 Charging for Application Signaling

Motivated by the previous DNS study, we also examine whether application signaling messages are also charged by 3G accounting and how much percentage they contribute when charged. Our findings show that, both 3G operators charge the signaling data (e.g., ICMP, SIP and RTSP) and protocol overhead. However, the actual signaling cost varies a lot across different applications. We consider three interesting cases.

FTP Signaling Channel: FTP uses two separate TCP sessions, with port 21 for command signaling and port 20 for data transfer. We conduct experiments to send messages mainly over the signaling channel (e.g., list a remote folder with 1 or 50 files), as well as data transfer (downloading one or ten 1MB files).

HTTP Redirect and Invalid Links: We run HTTP redirect cases, where the web page is redirected once or 15 times to reach the final content. We also access a web page with one invalid HTTP link. In the invalid link case, we access a web page that has one or 50 invalid image links.

Email and IM: We tested Yahoo Mail and Skype for Email and IM applications, respectively. We send a small or large email, login/logout skype, or remain idle for 10 minutes in Skype.

Results: The results are shown in Table 4. We make three observations. First, the application signaling messages, including FTP control commands, are indeed charged. Second, the signaling overhead percentage is particularly large when the content size is small, e.g., FTP listing, HTTP redirection. Command messages may only be a small percentage in the operator's charging volume, compared with protocol overheads (see FTP signaling); Third, signaling and protocol overhead do incur hidden costs (not perceived by average users). Note that, in the HTTP case, those invalid links or those redirects are never the content requested by the user. It explains why Alice is charged by an invalid click without accessing real content in the itemized bill of Figure 1.

Application	Test	Content (KB)	OP (KB)	Gap	Gap/OP (%)
FTP	Listing (1)	0.06	2.97	2.91	97.9
	Listing (50)	3.32	7.28	3.96	54.4
	Downloading (S)	1024	1190.5	166.5	14
	Downloading (L)	10240	10858.0	618.0	5.7
HTTP	Redirect (1)	0.05	1.9	1.85	97.3
	Redirect (15)	0.05	15.1	15.05	99.7
	Invalid (S)	0.13	2.05	1.92	93.6
	Invalid (L)	2.56	12	9.44	78.7
Email	Send (S)	0.02	13.0	12.98	99.8
	Send (L)	223.6	250.98	27.38	10.9
Skype	login/out	0	50.07	50.07	100
	idle (10mins)	0	5.05	5.05	100

Table 4: Signaling overhead of popular applications.

In a broader view, the above study makes us contemplate on who should pay for what. We have seen that free applications may raise more overhead for advertisements, while the paid ones may not. As the network moves toward content-based operations, lots of interesting issues and debates may arise.

7. DISCUSSIONS

We have described the accounting discrepancy in both extreme and common settings. The most optimistic view will claim that, the problem is not too bad, so we do not need to fix it; the built-in control mechanisms in TCP and applications at the end devices help to mitigate the damage. However, we believe that there are fundamental technical problems beneath these engineering missteps. We now discuss two issues: the architecture options and policy practice.

Rethinking Accounting Architecture: In general, there are three classes of accounting architecture: the network-based one such as the current 3G system, the end-system-based approach, and the collaborative one between the network and end devices.

Both the network-based and the end-system-based approaches have severe limitations. For the network-based, 3G charging system, we already observed that it should result in large accounting gap in the extreme cases. The fundamental problem is that, the network lacks coordination with end systems and makes the charging decision alone. This functions fine when everything goes well, but suffers when things go wrong. The built-in feedback loop in TCP and applications may help to certain extent. However, the concrete feedback mechanism and its operation accuracy are largely out of control to the accounting system. The charging system is not self-healing under failures and extreme conditions. On the other hand, the other extreme of end-system-based accounting will not work either. There is no easy mechanism to regulate users so that they will not cheat. The verification process of user-reported results is also challenging. Therefore, this approach is unrealistic in practice.

We believe that the coordinated charging system between the network and end systems is promising. The data delivery process is end to end, so both the network and end devices are players in data delivery. They need to make concerted decision in charging, too. Of course, there are various forms of coordination between the network and the device. We are not advocating the scheme that both parties play symmetric roles. Instead, we believe that the network has to take more central role in the charging process, while the end devices offer useful hints and feedback to the network. The network naturally has "centralized" views on users and flows, and more resources to control and regulate the charging decision. To build a more resilient charging system, several challenges arise along this direction: (1) What failures and losses does the accounting system have to handle? (2) What mechanisms are indispensable to coping with given failures? (3) When and how does the end device/server report delivery losses? (4) How does the system ensure that the feedback information is secure and trustworthy? (5)

How many mechanisms need to be placed into the future cellular network standards?

Policy as Double-Edged Sword: Policy practice is an inherent component of the accounting systems for mobile users. Policies can be good for both operators and end users! On one hand, policy practice offers carriers flexibility, while injecting dynamics into the market. It can serve as a viable mechanism to compete with other carriers when offering users better services at lower cost during certain times. On the other hand, users can also benefit from policy practice. As we have seen in the DNS case, users will pay less due to the free DNS service! However, we have to be prudent with policy practice. The policy choice needs to be conflict free. Moreover, its enforcement has to be strict. Otherwise, policy may open holes that operators may never anticipate.

8. RELATED WORK

Despite the popularity of 3G/4G data services, cellular network accounting remains a largely unaddressed area in the research community. [20] offers a nice survey on pricing, charging, billing methods for 3G systems in 2005. Among current industry efforts, Cisco proposed overcharging protection for GGSN to be aware of lost coverage at end devices based on the feedback of SGSN [1]. Using certain, free Access Point Name (APN) provides another way to obtain free data service in certain carriers, e.g., AirTel India [2] and UK Three [6]. DNS tunneling is also used in the iodine tool [4], designed for data access in scenarios different from ours, where DNS queries are allowed but the Internet access is blocked. In contrast, our work examines the current accounting practice in operational 3G networks. We use experiments to study various charging behaviors within the 3G accounting standards and policy practice by operators, identify root causes and propose fixes.

In the more general context, Internet accounting and pricing have been explored in the literature [19, 22, 28, 29] (see [24] for a survey for work up to 2001). These prior efforts focus on the wired Internet. The proposed accounting solutions are quite different from the one used by current cellular networks.

9. CONCLUSION

The Internet is going wireless and mobile. Two underlying driving forces have been the explosive growth of smartphones/tablets and the rapid deployment of 3G/4G infrastructure. Unlike the wired Internet, cellular networks have implemented usage-based charging, rather than the simpler flat-rate charging. Going down this path, the 3G/4G standards finalize the accounting architecture, yet leave enough freedom for operators to define their own charging policy. In this work, we conduct experiments on operational 3G networks to study the implication of such an architecture and practice, and quantify the charging discrepancy between the operator's record and the user's observed volume.

Our study offers some insights. On the architecture side, the SGSN/GGSN element-based charging is easy to implement, yet poses limitations. When things go wrong outside the charging elements, the resulting data volume deviates from what is observed at end devices. The fundamental problem is that, the charging system mainly uses open-loop, but not closed-loop operations; it makes accounting decisions alone without taking feedback from end devices. On the other hand, policy offers flexibility, but is also mistake prone. Policy operators have to take extra care to make policy enforcement complete. On both fronts, really bad things can happen under extreme conditions, somewhat unexpectedly. Consequently, as shown by our extensive experiments, we may pay for what we never get and get what we want for free in worst-case scenarios. We hope our initial efforts will stimulate further research on this important topic from both academia and industry.

Acknowledgment

We greatly appreciate the insightful and timely comments from our shepherd, Dr. Dina Papagiannaki, and the anonymous reviewers for their constructive feedback. We also thank Mr. Xingyu Ma and other participants in the charging experiments.

10. REFERENCES

[1] Cisco ASR 5000 Series Serving GPRS Support Node Administration Guide. `http://www.cisco.com/en/US/docs/wireless/asr_5000/12_0/OL-24828_SGSN_Admin.pdf`.
[2] Free GPRS Hack For Reliance Mobile. `http://www.megaleecher.net/Reliance_Internet_GPRS_Hack`.
[3] Free Gprs Mobile Tricks. http://darkwap.mobi/gprs-tricks/Free-Gprs-Mobile-Tricks.
[4] Iodine. http://code.kryo.se/iodine/.
[5] PPS. http://www.pps.tv/.
[6] Three PAYG Mobile Internet for FREE. http://www.digitalworldz.co.uk/226311-three-payg-mobile-internet.html.
[7] Traffic Monitor - RadioOpt GmbH. https://play.google.com/store/apps/details?id=com.radioopt.widget.
[8] TrafficStats. http://developer.android.com.
[9] VLC Stream & Convert. http://traveldevel.com/.
[10] Wing FTP. http://www.wftpserver.com/.
[11] 3GPP. TS23.060: GPRS; Service description; Stage 2, Dec. 2006.
[12] 3GPP. TS23.125: Overall High Level Functionality and Architecture Impacts of Flow Based Charging, Mar 2006.
[13] 3GPP. TS32.240:Telecommunication management; Charging management; Charging architecture and principles, Sep. 2006.
[14] 3GPP. TS25.301: Radio Interface Protocol Architecture, 2008.
[15] 3GPP. TS25.413: UTRAN Iu interface RANAP Signaling, 2008.
[16] 3GPP. TS23.401: GPRS Enhancements for E-UTRAN Access, 2011.
[17] G. America. Global 3G Deployments UMTS HSPA HSPA+, 2010.
[18] Caida. http://www.caida.org/research/traffic-analysis/tcpudpratio/.
[19] R. Cocchi, S. Shenker, D. Estrin, and L. Zhang. Pricing in Computer Networks: Motivation, Formulation, and Example. *IEEE/ACM Transactions on Networking*, 1:614–627, 1993.
[20] Z. Ezziane. Charging and Pricing Challenges for 3G systems. *IEEE Communications Surveys and Tutorials*, 7(1-4):58–68, 2005.
[21] H. Holma and A. Toskala. *LTE for UMTS: Evolution to LTE-Advanced*. Wiley, 2011.
[22] M. Kouadio and U. Pooch. A Aaxonomy and Design Considerations for Internet Accounting. *SIGCOMM Comput. Commun. Rev.*, 32(5):39–48, Nov. 2002.
[23] mobiThinking. Global Mobile Statistics 2012. http://mobithinking.com/mobile-marketing-tools/latest-mobile-stats.
[24] A. Odlyzko. Internet Pricing and the History of Communications. *Computer Networks*, 36:493–517, 2001.
[25] OECD. Nearly Two-Thirds of US Broadband Subscribers are Wireless. http://www.websiteoptimization.com/bw/1012/.
[26] D. M. Reporter. Up to 20 million Americans 'Overcharged' by AT&T for Data Usage, 2011.
[27] RFC5966: DNS Transport over TCP - Implementation Requirements, 2010.
[28] S. Shakkottai, R. Srikant, A. E. Ozdaglar, and D. Acemoglu. The Price of Simplicity. *IEEE Journal on Selected Areas in Communications*, 26(7):1269–1276, 2008.
[29] D. Trossen and G. Biczók. Not Paying the Truck Driver: Differentiated Pricing for the Future Internet. In *Proceedings of the Re-Architecting the Internet Workshop*, ReARCH '10, 2010.
[30] Z. Wang, Z. Qian, Q. Xu, Z. Mao, and M. Zhang. An Untold Story of Middleboxes in Cellular Networks. In *SIGCOMM '11*, 2011.
[31] WireShark. http://www.wireshark.org/.
[32] X. Zhang, Y. Xu, H. Hu, Y. Liu, Z. Guo, and Y. Wang. Profiling Skype Video Calls: Rate Control and Video Quality. In *INFOCOM'12*, March 2012.