Impact of Malicious TCP Retransmission on Cellular Traffic Accounting

Younghwan Go, Denis Foo Kune[†], Shinae Woo, KyoungSoo Park, and Yongdae Kim

KAIST University of Michigan[†]

ABSTRACT

Current cellular network architecture charges users based on the consumed IP traffic in byte-level. While the users receive payload at the application layer, there exists a transport layer (TCP) that can cause additional data consumption due to the retransmission that guarantees a reliable delivery. In this paper, we examine the accounting policies regarding the TCP's retransmission in five major cellular ISPs in the U.S. and South Korea and show that the current implementation is either vulnerable to billing inflation or allows the subscribers to bypass the charging. Two ISPs account for all retransmission packets, allowing attackers to inflate a victim's bill by intentionally retransmitting packets. Three ISPs exclude the retransmission packets from the user's bill thus allowing tunneling through TCP retransmissions. We present real-world attack scenarios where "usage-inflation" and "free-riding" attacks are plausible.

1. INTRODUCTION

Cellular 3G/4G data traffic is rapidly increasing. The volume is predicted to reach 10.8 Exabytes per month in 2016, which is an 18-fold increase from that of 2011 [1]. The number of cellular network users has already reached 1.2 billion worldwide [2], with the alarming growth rate of 64% in the U.S. and 85% in South Korea [3], and it is estimated that 85% of the world population will subscribe to the cellular network service by 2017 [4]. The explosive demand for cellular Internet access seems to persist at least for foreseeable future.

Given the increasing demand in the cellular traffic, accurate accounting of the traffic usage becomes all the more important. Most cellular ISPs adopt the pay-per-usage charging model for cellular Internet access. Subscribers typically buy a monthly usage plan (e.g., 2 GB per month) and the ISPs enforce it by byte-level accounting of the consumed IP packets. However, this approach presents an important policy decision for the TCP traffic. ISPs now need to decide whether they account for retransmitted TCP packets or not. If the ISPs reflect the retransmitted packets into the bill, it may be unfair to the users especially when the packet delay variance or losses are due to a poorly-provisioned infrastructure. The natural alternative is to remove the retransmitted packets from the bill, but accounting becomes expensive since it has to manage every TCP flow for each subscriber.

To better understand the current practice, we examine the accounting policies for TCP retransmission with five large cellular ISPs in the U.S. and South Korea. Surprisingly, we find that the accounting policies vary between ISPs. Our measurements reveal that two U.S. ISPs account for every packet regardless of TCP retransmission. Moreover, we find that the users in these ISPs can be the target of a "usage-inflation" attack that maliciously retransmits packets even if there is no packet loss. The three ISPs in South Korea intentionally remove the retransmitted amount from the usage statistics. However, we confirm that their implementation allows free data transfers if attackers tunnel their packets inside TCP retransmissions. In this work, we show that it is easy to launch "usage-inflation" or "free-riding" attacks and to take control over current cellular accounting systems.

2. BACKGROUND

In this section, we describe the basic architecture of 3G/4G cellular networks and their accounting process. We mainly focus on the Universal Mobile Telecommunications System (UMTS) [5] for 3G and Long Term Evolution (LTE) [6] for 4G. The architecture is based on a Packet-Switched (PS) domain, in which the data is transferred in packets [7, 8]. Although we mainly focus on 3G, similar argument can be made for the 4G system as well.

2.1 3G/4G Accounting System Architecture

In the UTMS/LTE cellular network architecture, the User Equipment (UE) communicates with a target server in the wired Internet by passing the packets through a Radio Access Network (RAN) and a Core Network (CN). The RAN is responsible for allowing wireless access to the UE and for providing a connection to its CN. After passing through the RAN, the packets from a UE enter the General Packet Radio Service (GPRS) through Serving GPRS Support Node (SGSN, S-GW for LTE), which is responsible for delivering packets to or from the UE within its service area. Then, the Gateway GPRS Support Node (GGSN, P-GW for LTE) sends these packets out to an external data network where the target server is located.

ISPs (Country)	Accounting Policy
AT&T, Verizon (U.S.)	All Packets
SKT, KT, LGU+ (South Korea)	Normal Packets

Table 1: Accounting policies for TCP retransmission

2.2 3G Accounting Process

The cellular data accounting is carried out inside the CN in the form of a Charging Data Record (CDR) via the serving nodes (SGSN, GGSN, S-GW, P-GW). The SGSN/S-GW collects the charging information related with the radio network usage while the GGSN/P-GW collects that of the external data network usage. While the UE downloads its requested content from the target server through the cellular network, the GSN/GWs record the traffic volume arriving to the CN in the form of IP packet. The accounting process continues until the communication is completed and the UE tears down the connection. When the session is finished, the CDRs stored in the GSN/GWs are forwarded to the Billing System (BS) via the Charging Gateway Function (CGF) and are processed to calculate the total data volume consumed by the particular session.

3. RETRANSMISSION EXPERIMENTS

In this section, we run various tests to figure out the accounting policies currently being enforced in current cellular ISPs (Table 1). We download a file from our custom Web server that manipulates the TCP packets and compare the accounted volume by the ISP and the byte count in the captured packet trace at the client. We use iPhone 4 (iOS 5.1.1) for AT&T, iPad 2 (iOS 5.1.1) for Verizon, and Galaxy S3 (Android 4.0.4) for SKT, KT and LGU+ as test clients.

3.1 Test Setup

To generate retransmission packets at will in the middle of a TCP connection, we build our own Web server. When a connection is established, the server opens a *raw* socket to read the IP packets from the client, delivers the requested content, and injects retransmission packets. For simplicity, our server maintains a TCP window size of one packet and does not implement congestion nor flow control. In the client side, we use *wget* to fetch the content from our server. For accurate verification of the accounting volume, we collect all packet traces at clients by running packet capture programs such as tcpdump [9] or pirni [10], and compare the byte count with the accounted number provided by each ISP. TCP packets for connection handshake and teardown, and other background traffic are carefully excluded from the results by subtracting them from the total value.

3.2 Experiment Results and Vulnerabilities

We run retransmission experiments to determine the accounting policies of various cellular service providers. Each test is run three times and we show the average value. The ISPs are addressed by number, with ISP-1, 2 and 3 based in South Korea and ISP-4 and 5 based in the U.S.

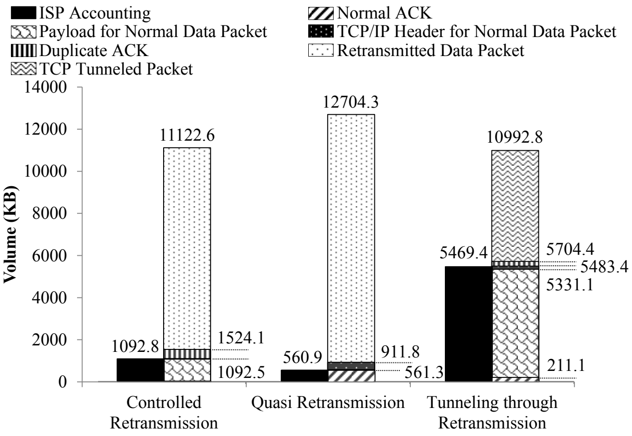


Figure 1: Experiment results of ISP-1 and 3

3.2.1 *Controlled Retransmissions*

In this experiment, we intentionally inject retransmission packets between each data packet. We initiate a TCP connection from the mobile client and then have the server send a pre-determined number of retransmission packets. We can easily calculate the total volume consumed, as the retransmissions will act as a simple multiplier. We test each ISP with 9 retransmissions per each data packet (e.g., a blowup by a factor of 10 in the real payload). We first discover that three ISPs in South Korea (ISP-1, 2 and 3) do not account for the retransmission packets. The two leftmost bars in Figure 1 and Figure 2 show the results for ISPs 1, 2 and 3. Interestingly, we see that the accounting policies for ISP-1 and 3 and ISP-2 are slightly different. Although they all ignore retransmitted data packets, ISP-2 accounts for duplicate ACKs while ISP-1 and 3 do not. We confirm that two U.S. ISPs count every retransmission, showing a blowup by a factor of 10 from the original file size. This test implies that the users in these ISPs can be the victim of usage-inflation attack.

3.2.2 *Quasi Retransmissions*

We look at how the service providers account for partial retransmissions where the next packet overlaps partially with the previous packet. For this, we send a small amount of application layer data (10KB, 75KB) for each ISP while the packet window is incremented by just one byte. We omit the ISPs that charge for retransmissions since they only account for the complete volume anyway. The two middle bars in Figure 1 show the result for ISP-1 and 3. We see that the ISP is not charging the TCP/IP headers for data with partially-retransmitted payload. On the contrary, ISP-2 (middle bars in Figure 2) accounts for all TCP/IP headers but not the retransmitted payload itself. This could be explained by an ISP that checks the sequence number and the packet length to identify the actual data volume but charges for the entire header since there is at least one byte of new payload.

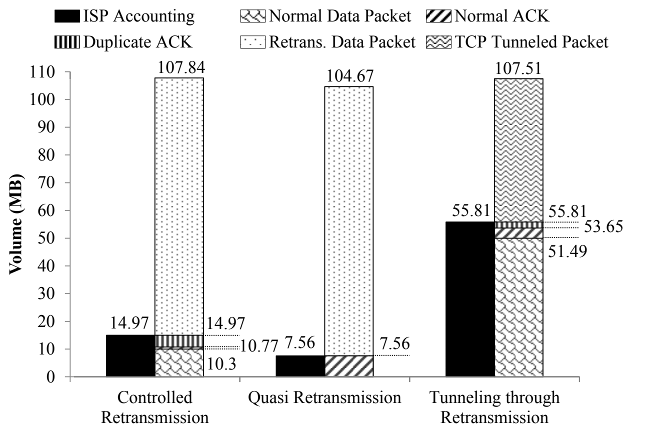


Figure 2: Experiment results of ISP-2

3.2.3 *Tunneling through Retransmissions*

Finally, we measure if the service providers verify that the data content of retransmissions do in fact contain a copy of the previous packet's payload data. If they only rely on the TCP headers, an attacker could set up a covert channel in the payload field of the TCP retransmission packets to avoid data charges. The two rightmost bars in Figures 1 and 2 show that all ISPs 1, 2 and 3 do not account for retransmitted packets with different payload. This makes intuitive sense since deep inspection of the TCP payload of every packet would be space and time consuming. From this test, we conclude that all ISPs that do not account for retransmitted packets are open to TCP-retransmission tunneling.

4. CELLULAR ACCOUNTING ATTACKS

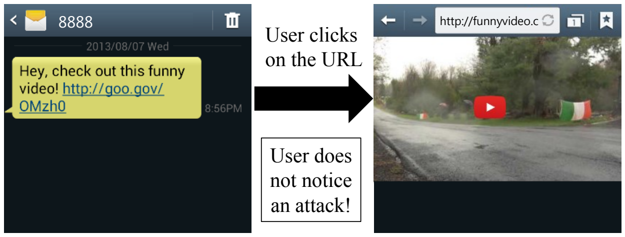
In this section, we look at possible design choices for "usage-inflation" and "free-riding" attacks by presenting real-world attack scenarios: phishing SMS and tunneling via proxy.

4.1 Usage-Inflation Attack

Figure 3 shows an attack scenario for "usage-inflation" attack. The attacker first sends a phishing SMS message to a target client with the URL that leads to a malicious site. When a user clicks on the link, she does not suspect any sign of attack since all she sees is an application layer content. However, the server begins to inject retransmission packets in the background, inflating the user's bill at the magnitude of the retransmission rate. This attack is easy to launch since it does not require compromising the client. As long as the user is redirected to a malicious server (via 3rd party advertisements, phishing emails or SMS messages), the attacker can inject any number of retransmission packets, which does not violate the TCP semantics.

4.2 Free-Riding Attack

Figure 4 shows a possible implementation of the "free-riding" attack. The "free-riding" attack requires a collaborating TCP tunneling proxy that relays the tunneled packets and real traffic between the client and the server. For downstream traffic, the packets from the server arrive at the proxy which tunnels them to the client, and the client de-tunnels and passes them to the application. In this architecture, the accounting system in the cellular core network will see only the connections between the client and the proxy with large numbers of retransmissions, which will not be counted in the final bill.



8888

2013/08/07 Wed

Hey, check out this funny video! http://goo.gov/OMzh0

8:56PM

Figure 3: "Usage-inflation" attack scenario

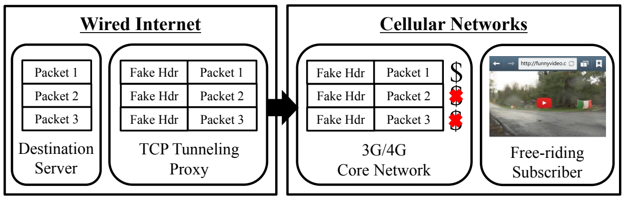


Figure 4: "Free-riding" attack scenario

5. CONCLUSION & FUTURE WORK

We have shown that due to the current design of the cellular data architecture and transport layer reliability mechanisms using retransmissions, the accounting policies either leave the user vulnerable to data inflation attack, or cause the ISP to be vulnerable to service charge evasion due to tunneling through retransmissions. For future work, we are implementing a practical attack framework, which exploits the current cellular infrastructure vulnerabilities. Moreover, we plan to build a cellular traffic monitoring system, which prevents the attacks, providing accurate billing to the subscribers.

6. REFERENCES

[1] CISCO. Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2011-2016. Technical report, 2012.

[2] ITU. ICT Facts and Figures. Technical report, 2011.

[3] M. Meeker and L. Wu. Internet Trends. Technical report, 2012.

[4] Ericsson. Traffic and Market Report. Technical report, 2012.

[5] 3GPP. UMTS. http://www.3gpp.org/Technologies/Keywords-Acronyms/article/umts.

[6] 3GPP. LTE. http://www.3gpp.org/LTE/.

[7] 3GPP. ETSI TS 132 200. 3rd Generation Partnership Project; Technical Specification Group Radio Access Network; UTRAN Iu Interface: general aspects and principles.

[8] 3GPP. ETSI TS 132 215. Telecommunication management; Charging management; Charging data description for the PS domain.

[9] Gadgetcat. tcpdump on Android, 2011. http://gadgetcat.wordpress.com/2011/09/11/tcpdump-on-android/.

[10] n1mda-dev. Pirni native iPhone ARP spoofer and network sniffer. http://code.google.com/p/n1mda-dev/wiki/PirniUsageGuide.