Splitting the Bill for Mobile Data with SIMlets

Himanshu Raj, Stefan Saroiu, Alec Wolman, Jitendra Padhye
Microsoft Research

Abstract: The scarcity of mobile broadband spectrum is a problem hurting all stakeholders in the mobile landscape – mobile operators (MOs), content providers, and mobile users. Building additional capacity is expensive, and MOs are reluctant to make such investments without a clear way of recouping their costs. This paper presents the idea of *split billing*: allowing content providers to pay for the traffic generated by mobile users visiting their websites or using their services. This creates an additional revenue stream for MOs and builds more pressure for updating their networks' capacities. End users also benefit because they can afford more expensive data plans and enjoy new applications and scenarios that make use of faster mobile networks.

To implement split billing securely on a mobile platform, we develop the *SIMlet*, a new trustworthy computing abstraction. A SIMlet can be bound to a network socket to monitor and account all the traffic exchanged over the network socket. SIMlets provide trustworthy proofs of a device's mobile traffic, and such proofs can be redeemed at a content provider involved in split billing.

Categories and Subject Descriptors

C.2.1 [**Network Architecture and Design**]: Wireless communication

General Terms

Design

Keywords

Mobile Computing, Traffic Metering, Traffic Billing, Broadband Wireless, 3G, 4G, ARM TrustZone, Split Billing, SIMlet

1. INTRODUCTION

The scarcity of spectrum for mobile broadband networks is hurting all stakeholders in the mobile landscape – users, content providers, and the mobile operators (MOs). As users migrate more of their computing tasks to smartphones and tablets using 3G/4G networks, users will increasingly feel constrained by the low capacity of today's mobile broadband networks. Content providers have little choice but to create "mobile-only" versions of their content to accommodate the lack of bandwidth reaching smartphones and tablets on the go. Finally, MOs are reluctant to make costly investments in upgrading their infrastructure. While MOs have access to a variety of approaches to improve their networks' capacity, all such approaches come with significant costs. Indeed, whether MOs purchase more spectrum, increase spatial reuse by deploying a higher density of cell towers, or deploy femtocells, MOs need strong incentives to invest billions of dollars in their infrastructure.

The current situation resembles a "chicken and egg" problem. MOs have little incentive to make expensive investments into their infrastructure as long as it is not clear how their costs will be recouped. Users are stuck with moderately expensive data plans, and are unwilling to pay more in the absence of compelling content and applications that need faster mobile broadband networks. Finally, content providers and application developers cannot "force" MOs to update their networks; instead, they alter their applications and services to fit current bandwidth rates and "spotty" reception.

There are only a few examples of content providers who have managed to overcome this situation. One example is Amazon's Kindle devices, which are sold with no need for customers to purchase data plans. Instead, Amazon bought data in bulk directly from AT&T. Amazon will recoup the cost of the 3G data as long as customers buy books from Amazon, and watch ads on their Kindles[1], all activities that consume relatively little bandwidth. Other examples are *0.facebook.com* [4], where Facebook offers free access to their site for all users of certain MOs, and Globe Telecom's *Free Zone*, a partnership between an MO in the Phillipines and Google [5]. Unfortunately, all such solutions face two challenges. First, they are heavy-handed because content providers and application developers must now sign deals with MOs; although Amazon can negotiate a deal with AT&T, it is much more difficult for the average app developer to create such a deal. The second challenge is that it is also expensive for MOs to negotiate individual deals with a large number of content providers. Such an approach to overcoming the "chicken and egg" problem does not scale.

This paper explores a different approach to overcoming this problem: *split billing*, a mechanism that allows content providers to subsidize or share the cost of 3G data consumed by their content and services. Split billing lowers the barriers between MOs, content providers, and users because it lets content providers to offer to pay for bandwidth on behalf of customers. This creates additional revenue streams for MOs, incentivizing them to upgrade their networks' capacities. End users also benefit because new scenarios are now possible, such as:

[1] Amazon offers cheaper versions of their devices that require users to watch ads. Ad-free Kindles are also available, but are more expensive.

HotMobile'13, February 26–27, 2013, Jekyll Island, Georgia, USA.

- Bandwidth-intensive websites such as Netflix or Hulu can enable mobile users to visit them and not worry about exceeding their bandwidth caps.

- Users can have a single phone for both work and personal use, and bill the network costs for their enterprise VPN traffic and other work-related applications directly to their employer.

- Advertisers can offer much richer ads (e.g., short video clips) to mobile users without having to worry about affecting their data plan quotas.

- Mobile phone users can run applications, such as peer-to-peer applications, where the traffic generated by the application is done on behalf of another user or device.

- Parents can encourage teenagers to run continuous monitoring applications to let them know where they are.

All these usage scenarios share a common restriction – a certain type of traffic meant for a certain destination or for use at a particular time should not be counted against the default quota.

Implementing split billing requires content providers and app developers to track the amount of 3G traffic their users consume. The system must have a high degree of security otherwise users might cheat in an attempt to qualify for savings on their 3G bill, by falsely claiming to have used 3G to access a site when in fact they used Wi-Fi. One possible implementation is to track users on the server-side and classify the type of network they are using (as 3G or Wi-Fi) based on the client's IP address. It is challenging to accurately infer which IP addresses are 3G versus Wi-Fi unless MOs cooperate with content providers and app developers. Section 4 describes the challenges of server-side split billing in more depth.

Instead, we chose an implementation that does not require cooperation from MOs: tracking 3G usage on the client-side. A client-side solution is more readily deployable because it does not require engagement with MOs. After a client-side approach has demonstrated the benefits of split billing, MOs will become more willing to take the necessary steps to offer split billing support to the masses of content providers and app developers. For example, MOs could offer a secure cloud-side service that tells app developers whether a given IP address is connected through their mobile broadband network. The app developer would use such a service to offer split billing inside their apps. Such a service must offer a high-degree of trustworthiness, otherwise mobile users could misrepresent their Wi-Fi traffic as mobile broadband traffic to the application's split billing component.

To implement client-side split billing, we develop an abstraction called a *SIMlet*. SIMlets are simple packet filters that associate individual traffic with a particular billing account. SIMlets must remain secure in the face of OS vulnerabilities, and thus they run in a strongly isolated runtime that leverages trusted computing hardware for mobile devices. Some of today's mobile phones already support dual SIM cards. SIMlets can be thought of as equivalent to being able to insert multiple SIM cards into a phone, along with a set of rules that indicate which particular SIM card a network flow should use.

Using SIMlets, we started the design and implementation of a fine-grained networking billing system for mobile devices where client applications or individual websites can create billing accounts and specify classes of traffic to be associated to a particular billing account. For each account, the system can provide trustworthy proofs of how much data applications have consumed. Users could redeem such proofs to MOs or content providers in exchange for rewards, rebates, credits, or even cash.

For our implementation, we use the Berkeley Packet Filter (BPF) to support the SIMlet abstraction, and we use the ARM TrustZone feature built into modern ARM System-on-Chip's (SoCs) to provide a trusted execution environment for SIMlets. Using TrustZone enables our system to perform trustworthy metering of which traffic is assigned to which billing account even if malware infects the OS running on the smartphone or tablet.

2. COST ANALYSIS

3G/4G spectrum crunch stems from the limited spectrum and long-range nature of the wireless technology. A single cell tower today handles 100's or even 1000's of individual subscribers at distances of up to tens of miles [13]. This is in sharp contrast to Wi-Fi where a one AP rarely handles more than tens of subscribers. With the unprecedented growth of smartphone and tablet usage, there simply is not enough network capacity to address the emerging demand. While we have seen a steady increase in the data rates supported by mobile broadband networks, the effective throughput for applications is substantially lower than the advertised rates [9], and we are already witnessing the effects of network congestion, with many users complaining of slow networks.

To address these capacity issues, network providers are considering a few options: 1) purchasing additional wireless spectrum, 2) increasing spatial reuse by deploying additional celltowers, and 3) increasing spatial reuse by deploying femtocells. While each of these options has the potential to improve the current capacity crunch, there is a common factor hindering all these options. They are each very expensive, and they come with significant deployment challenges. For example, increasing celltower density requires leasing or buying more physical locations, obtaining permits, and connecting all these sites, which increases fixed network costs and complicates operations.

For split billing to be viable, the customer base must be attractive to content providers and app developers. Content providers and app developers must be able to recoup the split billing costs and even sell more services to customers. For example, a movie download provider could offer free movie previews to entice its customers to purchase more movies. A mobile game developer might offer free 3G for its games as long as customers keep making in-app purchases. Split billing is attractive as long as content providers and app developers can target customers with the means to pay for additional content and services.

The remainder of this section describes a cost analysis of the feasibility of split billing. We combine three datasets to show that today's data plans come with strict quotas that provide little data to their users. The second high-level finding is that some well-developed countries offer worse data plans (by both cost and quantity of data) than some less-developed countries. This suggests that split billing could be effective in attracting customers more likely to spend on additional content and services.

2.1 Data Sources

The data plans used in our analysis come from a dataset of prices of mobile broadband released by Google[2]. Table 1 shows a few high-level statistics of this dataset. We also breakdown some of our analysis by countries and their GDP; the GDP data comes from

[2] http://policybythenumbers.blogspot.gr/2012/08/international-broadband-pricing-study.html

	Mobile Broadband Pricing (Google dataset)
Number of Plans Examined	2154
Number of Countries	106
Types of Mobile Broadband	2G, 2.5G, 3G, 4G, CDMA, EDGE, EVDO, HDSPA, HSDPA, HSPA, LTE
Number of Unlimited Plans	142 (4.9%)

Table 1: High-level statistics of the mobile broadband pricing data released by Google.

the World Bank[3]. Finally, we use estimates of the today's 3G/4G bandwidths collected by a very recent paper [16].

2.2 Today's Data Plans Offer Little Data

The pricing dataset revealed by Google shows that most data plans have very strict quotas. On the left, Figure 1 illustrates the distribution of monthly quotas. Almost 85% of all plans offer less than 10GB of data a month, and 36% offer less than 1GB a month. Even worse, while unlimited data plans which were common a few years ago have all but disappeared. In fact, we found less than 5% of plans to offer unlimited data (these plans were removed from the data plotted in Figure 1). Unfortunately, we cannot investigate the popularity of each data plan because the pricing dataset does not include the number of subscribers for each plan.

To better display how little data these plans offer, on the right, Figure 1 illustrates how many hours of downloads (or uploads) a 10GB plan offers given current 3G/4G bandwidth found in three major US cities: New York, Los Angeles, and Chicago. We use recent measurements of average download and upload speeds collected by [16]. In NY alone, a 10GB monthly data plan would be fully exhausted in less than 7 hours. Since uploads are slower, a 10GB plan could sustain about 12 hours of uploads a month. These findings show that today's plans offer little data to their users.

2.3 The Viability of Split Billing

To examine this issue in more depth, we perform the following experiment. We classify the countries present in the Google dataset in two ways: by their GDP per-capita in US$ and by the per gigabyte cost of their cheapest data plan. If some of the "richest" customers have access to inexpensive data plans then split billing is less attractive to them. Conversely, if data plans are very expensive only to those customers living in countries with low GDPs, these customers are also less likely to afford additional content and services.

Table 2 lists the top 10 "richest" countries in each category: by GDP in US$ and by per-GB cost. The two datasets are completely disjoint; seven of the top 10 "richest" countries by per-GB cost are located in Africa, a continent that has no country in the top 10 "richest" by GDP. Note that we only consider the countries listed in the Google dataset; for example, although Luxembourg has the highest per-capita GDP in the world, it is not listed in Table 2 because the Google dataset does not include any data plans from Luxembourg.

Even worse, unlimited plans are equally unavailable to customers in countries with high GDP as well as low GDP. Out of the top 20 countries with highest GDP, only seven of them have access to unlimited data plans. In contrast, six countries out of the bottom 20 also have access to unlimited data. All these results show that high-GDP customers do not benefit from cheaper plans and more data. All these findings suggest that split billing could be used effectively to attract this class of customers.

[3] http://data.worldbank.org/indicator/NY.GDP.PCAP.CD

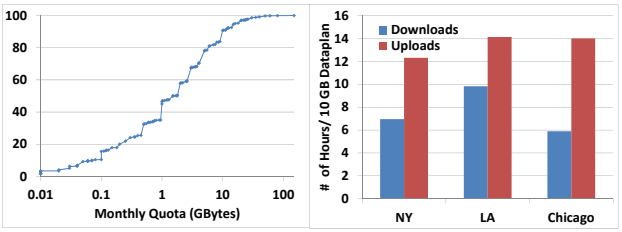

Figure 1: The distribution of monthly quotas for mobile broadband plans (left), and the number of hours of connectivity for monthly plans in three major US cities (right).

3. OUR SPLIT BILLING DESIGN

Split billing is the division of a customer's data plan bill into multiple parts. Our design deliberately avoids relying on MOs to make changes inside their networks. This design choice leads to a much quicker path to deploying split billing in the wild. After the benefits of split billing have been demonstrated, MOs can then offer different mechanisms that implement split billing support inside their network. Therefore, we start with the following four design goals:

1. *No changes inside the network.* Our system should be implemented by making changes to the endpoint mobile devices only. The design originated from our desire of quick prototyping. Requiring changes to the mobile network would drastically raise deployment costs and challenges.

2. *The security of split billing should be comparable to the security of SIM cards today*. It should be very difficult for users to bypass our security and charge for "bogus" traffic. Although impossible to offer "perfect security" (e.g., SIM cards can be lost and stolen, or subject to sophisticated physical attacks [7]), we aim to maintain similar levels of security as with today's SIM cards.

3. *Offer a form of accountability.* Anyone, whether a user, content provider, or mobile operator, should be able to verify who is responsible for the data plan charges.

4. *Provide adequate performance.* Any performance overhead should be negligible to users.

3.1 High-Level System Overview

In our system, each mobile device manages a set of rules on how to bill the 3G/4G data. For example, data consumed by visiting Netflix could be billed to the user's Netflix account, whereas data consumed over an enterprise VPN connection could be billed to the user's employer. Split billing runs strongly isolated from the rest of the system in such a way that any security compromise (including an OS compromise) will not compromise network metering. Periodically, our system produces a bill that tallies the 3G traffic consumed against each of the rules entered in the system.

We rely on the mobile device's trusted computing features (ARM TrustZone [3]) to isolate the split billing code from the rest of the running system and to produce an attestation for each bill (e.g., using a mechanism similar to remote attestation provided by TPMs [19]). ARM TrustZone enables this approach: it provides two runtime environments called the normal world and the secure world, with hardware support for isolation. This ensures that secure world memory is isolated from the normal world. The OS and all applications run in the normal world, and only the trusted components that implement split billing run in the secure world. A

	GDP Per-Capita	Cost per GByte
1.	Switzerland	Algeria
2.	Australia	Papua New Guinea
3.	Denmark	Cameroon
4.	Sweden	Iraq
5.	Canada	Angola
6.	Holland	Haiti
7.	Austria	Zimbabwe
8.	Finland	Chad
9.	United States	Mali
10.	Belgium	Sierra Leone

Table 2: Top 10 countries by GDP per-capita and cost per gigabyte.

more in-depth description of the ARM TrustZone technology can be found in [3].

This high-level description raises two challenges. First, what is a good abstraction for encapsulating the set of rules on a device that dictates how to split the the user's 3G bill? Second, how can we implement this abstraction securely by leveraging the TrustZone found on commodity ARM-based mobile devices?

3.2 Implementing Split Billing with SIMlets

To implement split billing, we develop a new abstraction, called a *SIMlet*. A SIMlet corresponds to a billing account, is issued by a content provider, and is destined to a smartphone user. A SIMlet also includes a *rule* that dictates the conditions under which 3G data is billed against the SIMlet. For example, Netflix could issue a SIMlet that specifies all 3G data to or from *netflix.com* is to be paid for by Netflix.

The issuer of a SIMlet is a content provider identified by its domain name and the recipient is a smartphone user identified by their phone number. A SIMlet has a start and an end date and can be used only during this time. We use Berkeley Packet Filter (BPF) syntax to specify the SIMlet's policy, although SIMlets could use a more expressive policy language if needed. Finally, SIMlets are signed by their issuers to prevent their modification.

3.3 Trusted SIMlet Manager

Each smartphone has a trusted SIMlet manager whose role is to request and store SIMlets. The SIMlet manager runs inside the TrustZone secure world and is therefore isolated from the smarthone's OS and applications. Any application can ask the SIMlet manager to request a SIMlet from a remote website. The SIMlet manager contacts the remote content provider, authenticates to it, and requests the SIMlet. The manager then stores the SIMlet locally and signals to the application whether the SIMlet request was successful or not. We envision two ways in which the trusted SIMlet manager can authenticate to a remote website, either using primitives provided by the physical SIM card or using TPM-like remote attestations.

Once it issues a SIMlet, a remote content provider commits to pay for a portion of the user's traffic. The provider is free to share this information with the MO if it wishes to do so. This could help with streamlining the customer's payment process, or it could form the basis for further negotiations between the MO and the content provider with respect to bandwidth prices.

3.4 Trusted Metering with SIMlets

Our system runs a trusted meter in the secure world whose role is to match network traffic against the appropriate SIMlets. For this, an application explicitly requests that the trusted meter bill its traffic against a particular SIMlet. The trusted meter instantiates a matching rule in a forwarding table that will match the application's traffic and meter it appropriately. To implement this approach, we use sockets and packet filters.

```
// find a simlet that can be used for our destination
IPSimlet ips = IpSimlet.DefaultSimlet;
foreach (IPSimlet tmp in SimletStorage) {
  if(tmp.Issuer.Contains("hulu.com")) {
    ips = tmp;
  }
}

// create socket
Socket s = new Socket(AddressFamily.InterNetwork,
                      SocketType.Stream, ProtocolType.TCP);
IPHostEntry ipHostInfo = Dns.Resolve("www.hulu.com");
IPAddress ipAddress = ipHostInfo.AddressList[0];
IPEndPoint ipe = new IPEndPoint(ipAddress, 80);

// bind the simlet to the socket
s.BindSimlet(ips);

//connect
try {
  s.connect()
} catch(SimletException sime) {
  Console.WriteLine("SimletException: {0}", sime.ToString());
} catch(SocketException socke) {
  Console.WriteLine("SocketException: {0}", socke.ToString());
} catch(Exception e) {
  Console.WriteLine("Unexpected exception: {0}", e.ToString());
}
```

Figure 2: C# code fragment for creating a socket and binding a SIMlet to it.

When opening a socket, an application requests a SIMlet that is then bound to the socket. This binding operation allows applications to decide which SIMlets they want to use, which ensures that the user has control of this process. Once a socket has been bound to a SIMlet, the smartphone OS will send all outgoing network traffic from the socket buffer to the TrustZone secure world. Once in the secure world, the SIMlet packet filter rules are evaluated for billing, and then the data is handed to the network stack. For incoming traffic, the network stack decides which packets are destined to which sockets, and then checks if the socket has a SIMlet bound to it. If so, the SIMlet is evaluated, and then the packets are handed off from the secure world back to the socket in the normal world. Figure 2 shows a C# example of how to create a TCP socket and bind a SIMlet from *hulu.com* to it.

4. SERVER-SIDE SPLIT BILLING

Another design alternative is to implement split billing on the server-side. The advantage of such a design is that it does not require any modifications to clients, and there is no need to leverage trusted computing for strong isolation. Servers are trustworthy because they are already under the control of the content provider, unlike mobile devices. However, a server-side architecture must solve the following two challenges.

First, content providers need to distinguish what type of network link (e.g. Wi-Fi vs. 3G) their customers use to connect to their services. This step must be done securely; a process relying on customers to *notify* servers when using 3G versus Wi-Fi could be abused. To qualify for rewards and subsidies, customers could cheat and claim they are using 3G when actually using Wi-Fi.

Instead, servers must classify IP addresses as 3G or Wi-Fi without the cooperation of clients. Since MOs control the space of IP addresses allocated to their 3G clients, one possibility is to ask the MOs to help with IP classification. Unfortunately, MOs do not currently advertise what IP address ranges they use for 3G, and, in our experience, they are reluctant to share this knowledge with app developers and content providers at large.

Another possibility is to use network measurements rather than

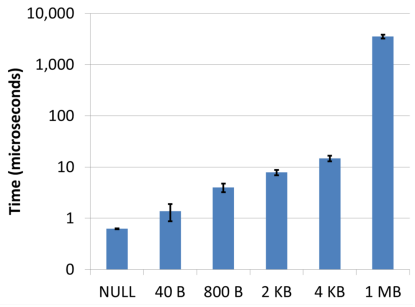

Figure 3: Overhead of RPC from normal world into secure one as a function of the size of RPC data.

MO cooperation to build catalogs of 3G IP addresses. Such a map can be constructed by combining a variety of techniques that gather information about IP addresses (e.g., reverse DNS lookup, autonomous system (AS) lookup, traceroute) and use various registries (e.g. RouteView [1], whois) to lookup information [17].

The second challenge is traffic counting. Server infrastructure is often geographically distributed, and sometimes these servers can be in different administrative domains under someone else's control, such as the case with content delivery networks (CDNs), or hosted servers (e.g., Microsoft's Azure or Amazon's AWS). It is often difficult to count traffic accurately across different servers placed in different organizations.

The major advantage of the server-side alternate design is that it requires no software deployment on the client, and thus there is no need for client-side trusted computing hardware. However, in the absence of cooperation from the MO, we think the problem of accurate IP address classification may be quite challenging to overcome. If IP classification is inaccurate, attackers will find ways to exploit it and cheat the split billing scheme.

5. PRELIMINARY EVALUATION

A key performance concern is the context-switching cost of transitioning from the normal world to secure world and back. To understand these costs, we performed a series of experiments in which the normal world executes a Remote Procedure Call (RPC) that transitions to the secure world, copies the RPC's arguments, and returns back a result. Such an experiment is a rough approximation of the split billing overhead as we must transition into the secure world each time data has to be metered.

Because all data sent through a socket with a SIMlet bound to it must be copied into the secure world, in our experiments we look at the cost of sending data of various sizes (40 bytes, 800 bytes, 2 KB, 4 KB, and 1 MB) in addition to performing null RPCs (i.e., no data copies). Each experiment is repeated 100 times and we present the average result and the standard deviation. Our experiments were done on an NVidia Harmony board with the Tegra 250 SoC. This SoC contains a dual core ARM Cortex A9 CPU running at 1 GHz with 1 GB of RAM.

Our results show that the cost of such transitions is very low. A null RPC call takes on average 624 nanoseconds with a standard deviation of 11.6 ns. Figure 3 shows the cost of an RPC call as a function of the size of the data copies. We find the cost of RPCs meets the performance needs of split billing. Even when copying 1MB of data to the secure world, the transitioning overhead due to metering is about 3.5 milliseconds.

6. RELATED WORK

A recent paper suggests that MOs should use time-dependent pricing of mobile data to manage their demand [8]. Similar to our work, this work's motivation stems from the huge growth in the demand for mobile data and the lack of capacity. However, charging more for data (which is what time-dependent pricing must do to reduce demand) makes mobile data plans even worse than they currently are. Instead, this work takes a different approach by engaging content providers and app developers in sharing the costs of mobile data.

Our system design borrows ideas from trusted computing hardware primitives to protect integrity and confidentiality for code and data [15, 12, 20, 11, 14]. Most of this previous work leverages the Trusted Platform Module (TPM) chips found on x86 hardware platforms [15, 12, 11]; few projects investigate the use of ARM's trusted computing primitives (ARM TrustZone) [10, 14, 20].

Our work has many parallels to the network neutrality debate and Section 7 examines this issue in more depth. Several previous projects have built and deployed techniques for quantifying the prevalence of traffic shaping on the Internet [18, 6], and recent work investigates the presence of traffic shaping in 3G networks [2].

7. DISCUSSION

This section presents three issues facing our implementation and the reasons why we remain optimistic that our system can overcome them. We do not claim to understand all issues our system will face in practice, nor that our way of addressing them will be the most effective. Our main goal is to obtain feedback at an early stage.

Split billing will increase the diversity of bandwidth pricing models which will in fact increase the incentives for traffic discrimination. It is true that split billing does not enforce the elimination of traffic discrimination. Large content providers can still negotiate preferential rates for their customer's traffic. However, one of the benefits provided by split billing is that it makes the network and traffic pricing more transparent. We believe that small content providers can only gain from increased transparency into data pricing and from access to a means for subsidizing their customers' bandwidth costs. Today's lack of transparency only hurts them, whereas our system democratizes access to billing agreements.

Mobile operators (MOs) fear of commoditization. Our system provides a simple way for content producers and users to enter agreements on who pays for bandwidth. One could argue that such agreements will make MOs feel that they have less control over the price of bandwidth and increase their fear of commoditization. Although we acknowledge that split billing might produce such an impression to MOs at first, we think MOs would embrace our system because it gives any content provider a simple way to pay for the bandwidth of its customers. Ultimately, MOs carry the traffic and thus can exert control over bandwidth prices. Also, split billing increases the pool of payees by including content providers (or any third-party) who may be willing to pay higher prices than consumers can tolerate.

Running a part of the billing system on a mobile device is too much of a security risk. This concern is serious – a security compromise of the TrustZone components of our system would allow an attacker to bill arbitrary 3G data traffic to any SIMlet. One way to mitigate this threat (other than the obvious way of strengthening the security of our system) is to build a way for content providers (or MOs) to audit the bills. For example, a content provider could also meter a customer's traffic on the server side and check whether

the bill presented by the customer matches the amount of traffic recorded at the server. Such audits do not have to run continuously and check all customers; instead, spot-checks would be able to determine whether our system provides adequate security.

8. CONCLUSIONS

This paper puts forward the idea of split billing: allowing content providers and app developers to pay for the traffic generated by mobile users when visiting their websites or using their services. We present a preliminary design of split billing on the client-side, without requiring any cooperation from the mobile operators. We believe that a client-side design offers a quick way to deploy split billing in the wild because it makes no assumptions about the network and takes no dependencies on the mobile operators.

To implement split billing securely on the client-side, we introduce a new trustworthy computing abstraction, called a SIMlet. We present a simple C# API on how SIMlets can be used by mobile applications. Finally, our evaluation shows that the overhead of copying data to and from the ARM TrustZone, an operation necessary for implementing SIMlets in a trustworthy manner, is low.

Acknowledgments

We would like to thank our shepherd, Suman Banerjee, and the anonymous reviewers for their valuable comments and feedback.

9. REFERENCES

[1] University of Oregon Route Views Project. http://www.routeviews.org/.

[2] WindRider: A Mobile Network Neutrality Monitoring System. http://www.cs.northwestern.edu/~ict992/mobile.htm.

[3] ARM Security Technology – Building a Secure System using TrustZone Technology. ARM Technical White Paper, 2005-2009.

[4] Fast and Free Facebook Mobile Access with *0.facebook.com*. http://www.facebook.com/blog/blog.php?post=391295167130, 2010.

[5] The Phillipines gets Facebook Zero-style free mobile access to Google services via Globe Telecom. http://thenextweb.com/mobile/2012/11/08/google-and-globe-telecom-launch-freezone-for-mobile-access-to-web-and-select-google-sites/, 2012.

[6] M. Dischinger, M. Marcon, S. Guha, K. P. Gummadi, R. Mahajan, and S. Saroiu. Glasnost: Enabling End Users to Detect Traffic Differentiation. In *Proc. of NSDI*, 2010.

[7] C. Gui, H. J. Wang, and W. Zhu. Smart-Phone Attacks and Defenses. In *Proc. of HotNets*, 2004.

[8] S. Ha, S. Sen, C. Joe-Wong, Y. Im, and M. Chiang. TUBE: Time-Dependent Pricing for Mobile Data. In *Proc. of Sigcomm*, 2012.

[9] J. Huang, Q. Xu, B. Tiwana, Z. M. Mao, M. Zhang, and P. Bahl. Anatomizing Application Performance Differences on Smartphones. In *Proc. of MobiSys*, 2010.

[10] H. Liu, S. Saroiu, A. Wolman, and H. Raj. Software Abstractions for Trusted Sensors. In *Proc. of MobiSys*, 2012.

[11] J. M. McCune, Y. Li, N. Qu, Z. Zhou, A. Datta, V. Gligor, and A. Perrig. TrustVisor: Efficient TCB Reduction and Attestation. In *Proc. of IEEE Symposium on Security and Privacy*, 2010.

[12] J. M. McCune, B. Parno, A. Perrig, M. K. Reiter, and H. Isozaki. Flicker: An Execution Infrastructure for TCB Minimization. In *Proc. of Eurosys*, 2008.

[13] RYSAVY Research. Mobile Broadband Capacity Constraints And the Need for Optimization. http://www.rysavy.com/Articles/2010_02_Rysavy_Mobile_Broadband_Capacity_Constraints.pdf, 2010.

[14] N. Santos, H. Raj, S. Saroiu, and A. Wolman. Trusted Language Runtime (TLR): Enabling Trusted Applications on Smartphones. In *Proc. of Hotmobile*, 2011.

[15] A. Seshadri, M. Luk, N. Qu, and A. Perrig. SecVisor: A Tiny Hypervisor to Provide Lifetime Kernel Code Integrity for Commodity OSes. In *Proc. of SOSP*, 2007.

[16] J. Sommers and P. Barford. Cell vs. WiFi: On the Performance of Metro Area Mobile Connections. In *Proc. of IMC*, 2012.

[17] N. Spring, R. Mahajan, and D. Wetherall. Measuring ISP Topologies with Rocketfuel. In *Proc. of SIGCOMM*, 2002.

[18] M. B. Tariq, M. Motiwala, N. Feamster, and M. Ammar. Detecting Network Neutrality Violations with Causal Inference. In *Proc. of CoNEXT*, 2009.

[19] Trusted Computing Group. PC client specific TPM interface specification (TIS), 2005.

[20] J. Winter. Trusted Computing Building Blocks for Embedded Linux-based ARM TrustZone Platforms. In *Proc. of STC*, 2008.