

Does Playing Online Mobile Games Overcharge You for the Fun?

De-Yu Chen
Department of Computer
Science, University of
California, Irvine
Irvine, CA, USA
teyuc@uci.edu

Po-Ching Lin
Department of Computer
Science and Information
Engineering, National Chung
Cheng University
Chiayi, Taiwan
pclin@cs.ccu.edu.tw

Kuan-Ta Chen
Institute of Information
Science, Academia Sinica
Taipei, Taiwan
swc@iis.sinica.edu.tw

ABSTRACT

With the growing popularity of online mobile games, the network traffic generated by them accounts for an increasingly significant proportion of mobile Internet traffic; however, the game's traffic characteristics have not been well studied. To understand more about such traffic, we analyze the network traffic of 9 online mobile games from 3 genres, namely, first-person shooting games, role-playing games, and racing games. Our results show that small payloads, high packet rates, and an excessive number of pure TCP ACK packets incur high traffic overheads. The payloads may have redundant content, which can be properly compressed. Given the increasing number of pay-by-volume and throttling 3G/4G data plans, gamers may not realize they are being overcharged due to such overheads. We observe that playing certain games costs players more than US\$5 per hour under certain data plans. We also find that the overheads can be reduced without compromising users' gaming experience because the packet payloads are highly redundant.

Categories and Subject Descriptors

C.2.2 [COMPUTER-COMMUNICATION NETWORKS]: Network Protocols—*Applications (SMTP, FTP, etc.)*

General Terms

Computer Networks

Keywords

online mobile games; network traffic; QoE

1. INTRODUCTION

Online mobile games are computer games played on mobile devices, such as smartphones and tablets, via the Internet. They have been extremely popular, thanks to the rapid

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or to publish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

MobiGames'15, May 19, 2015, Florence, Italy.

Copyright is held by the owner/author(s). Publication rights licensed to ACM.

ACM 978-1-4503-3499-0/15/05 ...\$15.00.

<http://dx.doi.org/10.1145/2751496.2751499>.

growth of mobile devices. Recently, Nouch [11] reported that the number of mobile gamers surpassed the 500 million mark in 2012, and they spent more than US\$9 billions. Khalaf [8] found that Android and iOS users spend 32% of their time in playing mobile games, which rank first among all types of mobile applications. Network traffic from mobile devices now accounts for a significant proportion of Internet traffic. A recent survey [5] found that global mobile network traffic amounted to 885 petabytes per month at the end of 2012; and it is expected to increase 13-fold by 2017. As the number of online mobile gamers grows rapidly, the network traffic generated by them will increase significantly [13]. Meanwhile, flat-rate 3G/4G data plans are being replaced by pay-by-volume or "throttling" data plan [7]. That is, when the volume of data is below a certain threshold, full bandwidth is provided; otherwise, a throttled bandwidth will be provided for the rest of the billing period. Thus, careful accounting of the network usage of mobile games is required.

Most network game traffic is *ad hoc* and highly dependent on the design and implementation of each game because there is no standard library, middleware, or transport protocol for online games. Moreover, fast-paced games, such as FPS, tend to use UDP instead of TCP as the transport protocol because the latter is regarded as less suitable for real-time gaming [2]. Therefore, the traffic characteristics of online games should be quite distinct from those of other popular network applications, and characterizing online mobile game traffic is an important research issue.

Currently, it is not known *if playing online mobile games overcharges players by sending out more traffic than necessary in order to overcome the uncertainty inherent in network transmissions*. Our objective is to determine whether current online mobile games have been carefully designed and handle network communications adequately. An inappropriate application design may cause 1) players to be overcharged due to excessive bandwidth usage, and 2) less than satisfactory game QoE due to too few updates (too conservative) or unnecessary congestion (too aggressive).

In this paper, we study 9 online mobile games from 3 popular genres, namely, first-person shooting (FPS) games, role-playing games (RPGs) and racing games (RCGs), on Apple's App Store (store.apple.com). We make the following observations based on the results:

1. The payload size of online mobile game traffic is generally rather small, which implies that the overheads of packet headers are relatively high.

2. Certain games, particularly those over UDP, have high packet rates, while some TCP-based games generate an excessive number of pure TCP ACK packets. Both types of games are associated with high packet header overheads.
3. We summarize the cost per hour of the 9 games based on the measured bitrates of the games according to the data plans offered by selected cellular Internet access providers.
4. Some games constantly send highly redundant packets that could be aggregated and compressed without compromising the gaming experience.

The remainder of this paper is organized as follows. Section 2 provides a review of related works. In Section 3, we describe the measurement setup for collecting the game traffic traces; in Section 4, we inspect the characteristics of mobile gaming traffic, such as header overhead; and, in Section 5, we investigate whether playing online mobile games over-charges players and how the problem can be remedied. Section 6 contains some concluding remarks.

2. RELATED WORK

According to Suznjevic and Matijasevic [12], research on the characteristics of online game traffic tries to improve the scalability, efficiency and quality of service of games in order to maximize gamers’ experience. They noted that the common characteristics of interest are the packet rate, payload size, bandwidth usage, and packet inter-arrival time. Chen et al. [4] observed that the payloads of ShenZhou Online, a Massive Multiplayer Online Role Playing Game (MMORPG), are rather small, particularly in traffic sent from clients to the server. The low packet rate leads to a relatively high ratio of pure TCP ACK packets because, normally, no more packets arrive before the expiry of the delayed ACK timer, which is usually set at 200 ms. The observations imply that the overheads of packet headers are relatively high in MMORPGs. Kim et al. [10] found that client packets in Lineage are quite small and narrowly distributed, whereas server packets are distributed more widely. Lineage II has highly asymmetric upstream traffic and downstream traffic; the latter requires much more bandwidth than the former [9].

While there have been a number of studies on the traffic analysis of online games, to the best of our knowledge, none have been dedicated to studying the traffic of online mobile games and the unique implications associated with mobile game traffic, such as data charging.

3. MEASUREMENT SETUP

We selected 9 mobile games from 3 popular genres (FPS, RPG and RCG) for analysis. All the games (see Table 1) can be played on Apple iPad 2 and freely downloaded from Apple’s App Store (as of June 2013). Since they are on-line games, they can only be played whenever an Internet connection is available. Among the 9 games, the 3 RPGs send packets over TCP connections, while the rest (3 FPS games and 3 RCGs) use UDP as their transport protocol. This split is basically consistent with the following suggestions in an FAQ article on GameDev.net [6]: “slow-paced games (e.g., RPGs) can use TCP, while fast-paced games (e.g., FPS games and RCGs) should use UDP.”

We connected an Apple iPad 2 to a wireless AP, which is provided by a Wi-Fi capable PC with an Intel i7 920

CPU and Windows 7, and ran `tcpdump` on it to capture all the packets flowing through the computer. We asked five gamers to play each of the 9 games for more than 10 minutes, and recorded the game traffic flowing through the Wi-Fi AP. The time spent in non-game-play intervals (e.g., the login screen, the chat room, and the player matching phases) were excluded from the trace. In total, we have collected 458 minutes of trace, which comprises 756 millions of packets and 95.8 GB of data.

4. TRAFFIC ANALYSIS

In this section, we investigate the mobile game traffic in a number of aspects. We inspect their traffic patterns and examine the source of high packet header overhead. We conclude this section with an empirical estimation of the games’ network bandwidth charges on the players. For brevity, we refer to the network traffic sent from the client to the server as *client traffic*, and that sent from the server to the client as *server traffic*.

4.1 Traffic Patterns

Figure 1(a) shows the cumulative distribution functions (CDFs) of the payload sizes (i.e., the packet sizes excluding the TCP/UDP/IP headers), the packet rates, and the bitrates for the client traffic in each game. Except `sr`, all the games have rather small payloads, i.e., most of them are 50 bytes or even smaller. While `bia2`, `ah7`, `sr`, and `oc` have slightly larger payloads in the server traffic, as shown in Figure 1(b), the other games have small payloads in both directions of traffic.

The client packet rates of `bia2` are generally higher than 80 pkts/sec, while those of the rest two FPS games are higher than 20 pkts/sec. Among the RPGs, `p1` has the highest packet rates as it generates 30 packets per second for most of time. The packet rates of the FPS games in the server traffic are higher than those in the client traffic, where the server packet rates of `bia2` and `sg` can be even higher than 100 pkts/sec. In Section 5, we will investigate whether such extremely high packet rates are really necessary for smooth gaming.

Given the small payloads, the overheads of the TCP/UDP/IP headers (i.e., the total header length divided by the total packet length) are relatively high. Figure 2 shows the header overheads for each game. The rates indicate that the overheads are generally high due to two factors: 1) the small packets and 2) TCP pure ACK packets. Note that a large number of client payload sizes are zero in the RPGs. These packets are pure TCP ACK packets, which simply acknowledge TCP payloads and do not carry any application data [3]. The pure ACK packets further increase the header overheads in the traffic of RPGs (as they use TCP to be the underlying transport protocol).

4.2 Pure TCP ACK packets

Figure 3 shows the ratios of pure TCP ACK packets in the RPG traffic. The server pure ACK packet rate is defined by the number of pure ACK packets sent *to* the server divided by the number of data packets (i.e., packets with a non-empty payload) sent *by* the server. The client pure ACK packet rate is defined similarly. According to the graph, almost every server data packet of `p1` triggers a pure ACK packet, which implies that 1) the delayed ACK mechanism [1] on the client is ineffective; and 2) a prominently

Table 1: The 9 online mobile games evaluated in this paper

| Category | Abbv. | Full game title | Release date | Publisher | Developer |
|----------|-------|-------------------------------|--------------|-------------------|-------------------|
| FPS | ar | Archetype HD | Aug 19, 2010 | Villain | Munkyfun |
| | bia2 | Brothers In Arms 2 Free+ | Feb 28, 2010 | Gameloft | Gameloft |
| | sg | SHADOWGUN | Sep 28, 2011 | Madfinger Games | Madfinger Games |
| RCG | ah7 | Asphalt 7: Heat | Sep 19, 2012 | Gameloft | Gameloft |
| | rr2 | Real Racing 2 | Dec 16, 2010 | Electronic Arts | Firemint |
| | sr | Sonic & SEGA All-Stars Racing | Feb 23, 2010 | SEGA | Sumo Digital |
| RPG | oc | Order & Chaos Online | Oct 31, 2011 | Gameloft | Gameloft |
| | pl | Pocket Legends | Apr 2, 2010 | Spacetime Studios | Spacetime Studios |
| | wom | The World of Magic | Apr 15, 2010 | Com2uS | Com2uS |

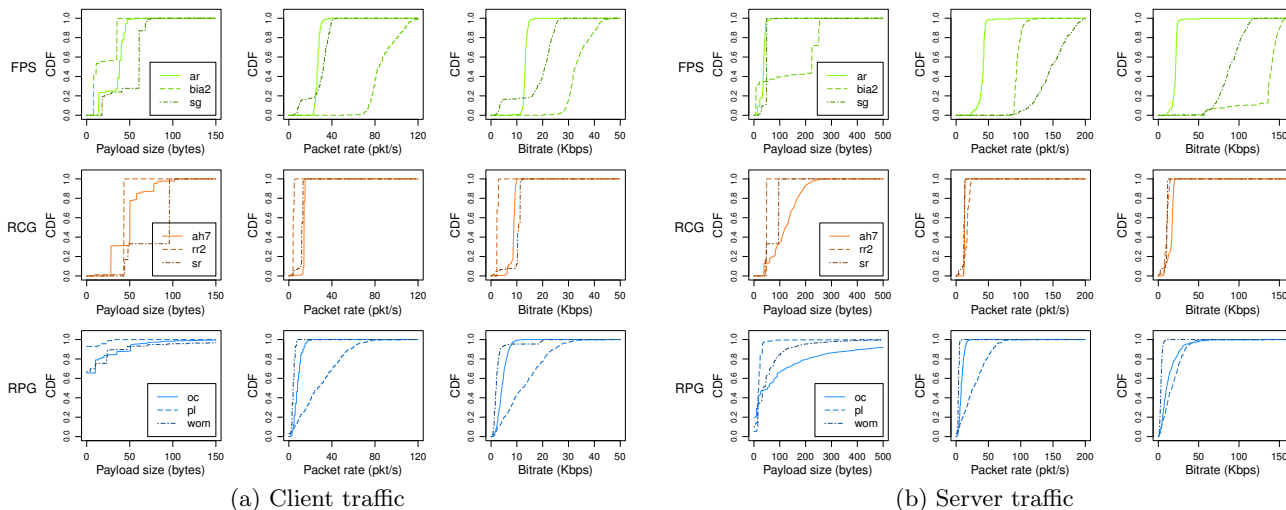


Figure 1: The CDF of the packet sizes, packet rates, and bitrates of observed game traffic.

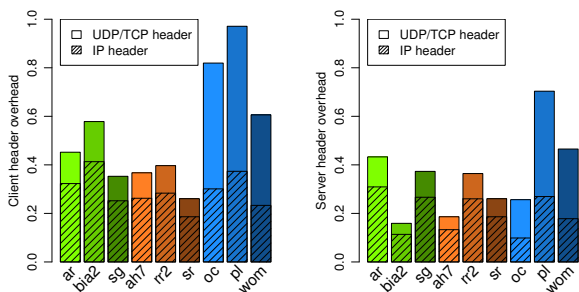


Figure 2: The TCP/UDP/IP header overhead in the game traffic

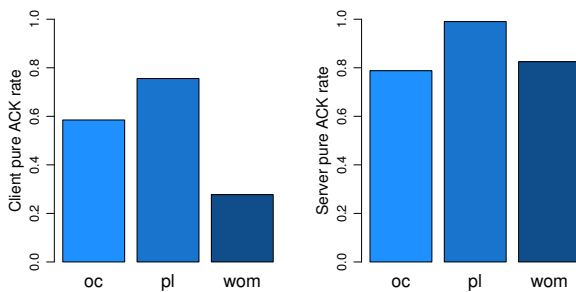


Figure 3: The rates of pure TCP ACK packets in RPG traffic.

large traffic overhead due to packet headers (c.f., Figure 2). The phenomenon also occurs in the other two RPGs, though less evident. As shown in Figure 1, the RPG packet rates in both directions are similar; however, most of the client packets are pure ACK packets and most of the server packets are data packets.

To verify our conjecture about the disabled delay ACK mechanism on the client, we designed a pair of simple TCP client/server programs and conducted an experiment. In the experiment, the server program is running on a Linux and the client program is on an Apple iPad2 in the same LAN. The server program sends out a one-byte message to the client every 1 second and 5 seconds respectively, and we record all the communications between both nodes using

`tcpdump`. Then, `tcptrace` was used to compute how much time was elapsed before an ACK packet was generated in response to a new data packet [1]. We call the elapsed time as *ACK delay time*. The distributions of the ACK delay time from our simple TCP client/server programs and those from the RPG traffic traces are presented in Figure 4. As can be seen, the ACK delay time of the custom TCP client/server programs are uniformly distributed between 0 and 100 ms, which manifests a delayed ACK timer of 100 ms [1]. In contrast, the ACK delay time of *oc* and *pl* are extremely short with most of the delay time nearly 0 ms, while those of *wom* show a mixture behavior—most of the pure ACK packets were sent with a 100-ms delay, but occasionally no ACK delay is present. The observations indicate that the

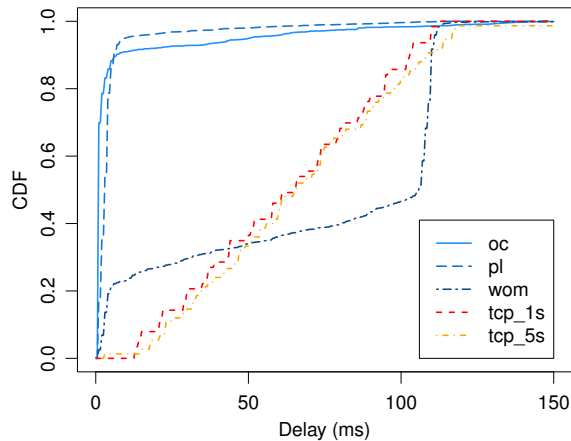


Figure 4: The distribution of the ACK delay time for 3 RPGs and our custom client/server programs

delayed ACK mechanism on iPad 2 is enabled by default, but mobile games can disable or even change the mechanism. We plan to look into this issue in the future.

4.3 Online Mobile Gaming Can Be Expensive

Under the pay-by-volume and throttling data plans now offered by major cellular Internet access providers, the high bitrates and high packet rates (and therefore the high traffic overhead) incur huge costs that will be passed on to gamers without their knowledge. Table 2 summarizes the cost per hour of the 9 games based on the measured bitrates of the games according to the data plan listed in Table 3. We observe that playing certain games, such as *bia2* and *sg*, costs players more than US\$5 per hour under certain data plans; and under the T-Mobile data plan in the UK, 8 out of the 9 games cost players more than US\$1 per hour. These high charges make us ponder: *Are the high packet rates and bitrates for gaming really necessary?* We will address this issue in the next section.

5. ARE ONLINE MOBILE GAMERS OVERCHARGED?

According to Figure 1, we can see that a considerable disparity exists between the bitrates and packet rates of the games that belong to the same genre. For example, the average bitrates of the FPS games (*ar*, *sg*, and *bia2*) are 20, 80, and 140 Kbps respectively, which correspond to ratios of 1:4:7 in their bitrates. It is undoubtedly that the communication requirements for each game can be very different due to differences in game design, the complexity of game scenes, the granularity of player actions, etc; thus, simply comparing the bandwidth usage of two games (even in the same genre) without considering those factors would be meaningless. However, given the surprisingly large difference in bitrates, in this section, we will investigate whether those online mobile games send more traffic than necessary and playing them overcharges the gamers.

5.1 Payload Can Be Less Redundant

We find that an online mobile game, if not well tuned, tends to send more data than necessary in order to han-

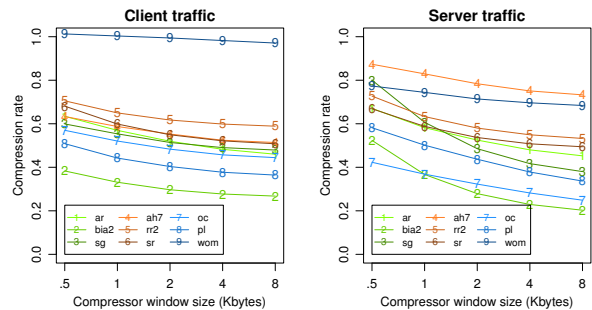


Figure 5: The compression rate of payload streams with various compressor window sizes

dle potential network impairments. In this case, the game’s payloads would contain a high degree of *redundancy* if the payload is not encrypted or obfuscated before transmission. According to this rationale, we extract and concatenate the payload stream of each game, and use *gzip* with different-sized compression windows to compress the payloads in order to compute the *compressibility* of the stream. Figure 5 shows the compression rates (derived by dividing the size of the compressed payload by the payload’s original size) for both client and server traffic with various window sizes. A lower compression rate indicates a higher compressibility and thus a higher degree of redundancy. A side-by-side examination of Figure 1 and Figure 5 reveals that, generally, the games with higher packet rates tend to have a higher degree of redundancy. For example, *bia2* has a high packet rate and a low compression rate, while the three RCGs and *wom* have low packet rates and high compression rates (i.e., lower redundancy). To illustrate the relationship more clearly, we plot the compression rates versus the packet rates and bitrates of the games in Figure 6. The graph shows that the payload compressibility is generally proportional to the packet rates and bitrates of the game traffic. This evidences that a game with a higher packet rate and bitrate may not be due to a more complicated game scene; the reason might simply be that *more redundancy exist in its traffic*. Further, the compressibility would make sense if the packets can be aggregated.

5.2 Packet Rate Can Be Lower

From another aspect, we also wonder whether a high packet rate is necessary for real-time online mobile gaming. To address this issue, we chose three games with higher packet rates, namely, *bia2*, *sg*, *pl*, and conducted a user study to empirically investigate whether the packet rates could be reduced without harming the user experience. A modified version of *dummynet*, which emulates a designated packet inter-arrival time for the server traffic, was run on the Wi-Fi AP. That is, the packets sent from the server to the client would be aggregated in the *dummynet* queue and released every t ms. Ten experienced mobile gamers were recruited to play the 3 games with t set between 0 ms (no aggregation) and 400 ms (corresponding to 2.5 pkts/sec), and we observed whether they perceived any negative effect due to the lower packet rates. More specifically, the subjects were asked to report their perceived level of “lags” in each round on a five-level scale (i.e., no delay, mild delay, moderate delay, severe delay, and very severe delay). Figure 7 presents the levels of perceived lags by the subjects. According to the graph,

Table 2: The estimated network cost per hour for each game.

| | | Japan | US | | | UK | |
|-----|------|--------|--------|----------|---------|----------|----------|
| | | DoCoMo | AT&T | T-Mobile | Verizon | T-Mobile | Vodafone |
| FPS | ar | \$0.78 | \$0.89 | \$2.13 | \$1.55 | \$2.37 | \$1.69 |
| | bia2 | \$5.06 | \$5.78 | \$13.87 | \$10.11 | \$15.44 | \$11.03 |
| | sg | \$3.22 | \$3.68 | \$8.83 | \$6.44 | \$9.83 | \$7.03 |
| RCG | ah7 | \$0.73 | \$0.83 | \$2.00 | \$1.46 | \$2.22 | \$1.59 |
| | rr2 | \$0.39 | \$0.44 | \$1.06 | \$0.77 | \$1.18 | \$0.85 |
| | sr | \$0.61 | \$0.70 | \$1.67 | \$1.22 | \$1.86 | \$1.33 |
| RPG | oc | \$0.55 | \$0.63 | \$1.51 | \$1.10 | \$1.69 | \$1.20 |
| | pl | \$0.93 | \$1.07 | \$2.56 | \$1.87 | \$2.85 | \$2.04 |
| | wom | \$0.19 | \$0.21 | \$0.52 | \$0.38 | \$0.57 | \$0.41 |

Unit: USD / hour

Table 3: The data plans offered by selected cellular Internet access providers.

| Company | Data plan | Monthly price (USD) [†] | Data included per month (MB) | Rate (USD/100MB) |
|--------------------|--------------------|----------------------------------|------------------------------|------------------|
| NTT DoCoMo (Japan) | Xi Pake-hodai Flat | \$61.26 | 7,168 | \$0.8546 |
| AT&T (US) | Plan for Tablets | \$50.00 | 5,120 | \$0.9766 |
| T-Mobile (US) | Simple Choice | \$60.00 | 2,560 | \$2.3438 |
| Verizon (US) | Share Everything | \$70.00 | 4,096 | \$1.7090 |
| T-Mobile (UK) | Pay Monthly | \$46.60 | 1,786 | \$2.6092 |
| Vodafone (UK) | Pay Monthly | \$38.20 | 2,048 | \$1.8652 |

[†] 1 British pound = 1.5279 U.S. dollars; 1 Japanese yen = 0.010235 U.S. dollars

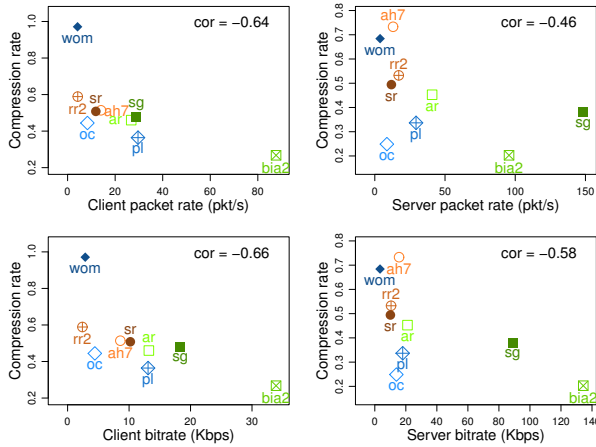


Figure 6: The packet rates / bitrates of games vs. the payload stream compression rates (8 KB window size)

aggregation delays up to 100 ms do not degrade the gaming experience of most players. That is, a game’s packet rates can be safely reduced to 10 pkts/sec without significantly degrading the gaming experience. More conservatively, a game can adopt a 50-ms aggregation delay (equivalent to 20 pkts/sec) that would yield significant reductions in packet rate and bitrate without sacrificing the QoE of the gamers.

Given the above results, we consider that the packet rates of the 3 FPS games we studied send out server packets with unnecessarily high packet rates; more specifically, **ar**, **bia2**, and **sg** can send 20 pkts/sec without QoE degradation instead of sending out 40, 100, and 150 packets per second on average and making their players overcharged with excessive bandwidth usage.

5.3 A Remedy Proposal

Given that online mobile games tend to send redundant traffic in an unnecessarily high packet rate, we propose to

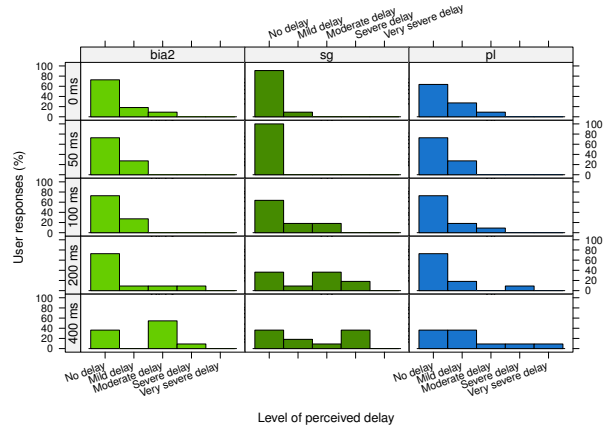


Figure 7: The perceived level of latency when playing games with packet aggregation

use a traffic optimization approach (i.e., aggregation plus compression) to resolve this issue without modifying the network engine of the games. The advantage of this approach is that it shifts the traffic optimization workload from the usually overloaded game developers to network programmers. Practically, the traffic optimization can be implemented in a socket-wrapper library, where the aggregation and compression are applied to any data to be sent by socket calls, and the reverse functions are applied to any data that is just received from another peer.

We verify the efficacy of the traffic optimization approach via simulations. In our simulations, the aggregation intervals are set to 100 ms as this seem to be an acceptable setting to most players (c.f., Figure 7). Rather than using a fixed window size as in Figure 5, the compressor now invokes **gzip** to compress the accumulated payload in the **dumynet** queue right before a packet is being released; also, we enable **gzip** to accumulate the dictionary across successive compression operations so that it can simply reuse symbols that have

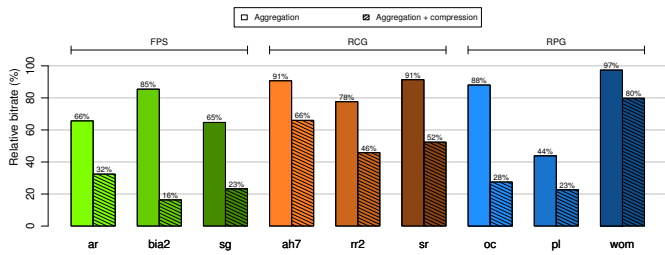


Figure 8: The relative server bitrate assuming aggregation (with 100 ms intervals) and a gzip compression algorithm is used

been stored for earlier packets. Figure 8 shows the relative bitrates (relative to the original bitrate without optimization) of the server traffic with aggregation and compression. We observe that even only with aggregation, the bitrate can be reduced to 45% to 80% on average. When aggregation and compression are both applied, the effect is significant—the bitrates of the FPS games can be reduced to 16% to 32% of the original bitrates; on average the games’ bitrates can be saved by around 50%. The worst case occurs for *wom*, which is expectable as it use of network bandwidth is quite conservative (with an average of 5 Kbps).

6. CONCLUSIONS

The gaming experience is visible to players, but the network bandwidth charges are relatively “invisible” to them. Therefore, online mobile game developers tend to overuse network bandwidth to ensure the network quality of online gaming regardless the overcharges for 3G/4G data plans paid by the gamers. In this work, we have shown that some online mobile games send more than necessary game traffic, which would cause problems for game players and the mobile Internet. We propose a remedy solution to optimize gaming traffic without changing the cores of the games; the simulation results show that the solution can save up to 88% of bandwidth charges for games without sacrificing gaming experience.

7. ACKNOWLEDGMENTS

This study is conducted under the Ministry of Science and Technology (MOST) Project 103-2221-E-001-023-MY2.

8. REFERENCES

- [1] R. Braden. Requirements for Internet hosts-communication layers, 1989. RFC 1122 (Standard).
- [2] K.-T. Chen, C.-Y. Huang, P. Huang, and C.-L. Lei. An empirical evaluation of TCP performance in online games. In *Proceedings of ACM SIGCHI ACE’06*, Los Angeles, USA, Jun 2006.
- [3] K.-T. Chen, C.-Y. Huang, P. Huang, and C.-L. Lei. An empirical evaluation of TCP performance in online games. In *Proceedings of ACM Advances in Computer Entertainment Technology (ACE)*, June 2006.
- [4] K.-T. Chen, P. Huang, and C.-L. Lei. Game traffic analysis: An MMORPG perspective. *Computer Networks*, 50(16):3002–3023, November 2006.

- [5] Cisco. Cisco visual networking index: Global mobile data traffic forecast update, 2012–2017, 2012. http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/white_paper_c11-520862.html.
- [6] gamedev.net. Networking and multiplayer FAQ, 2004. <http://www.gamedev.net/index.php?app=forums&module=forums§ion=rules&f=15>.
- [7] W. Gordon. The death of unlimited data: What it means, and how you can keep your unlimited data plan, 2012. <http://lifehacker.com/5917843/the-death-of-unlimited-data-what-it-means-and-how-you-can-keep-your-unlimited-data-plan>.
- [8] S. Khalaf. Flurry five-year report: It’s an app world. the web just lives in it, 2013. <http://blog.flurry.com/bid/95723/Flurry-Five-Year-Report-It-s-an-App-World-The-Just-Web-Lives-in-It>.
- [9] J. Kim, J. Choi, D. Chang, T. Kwon, Y. Choi, and E. Yuk. Traffic characteristics of a massively multi-player online role playing game. In *Proceedings of ACM NetGames 2003*, October 2003.
- [10] J. Kim, E. Hong, and Y. Choi. Measurement and analysis of a massively multiplayer online role playing game traffic. In *Proceedings of Advanced Network Conference*, August 2003.
- [11] J. Nouch. Mobile games market grew 33 percent in 2012 to \$9 billion, 2013. <http://www.pocketgamer.biz/r/PG.Biz/Newzoo+news/news.asp?c=48541>.
- [12] M. Suznjevic and M. Matijasevic. Player behavior and traffic characterization for MMORPGs: a survey. *Computer Networks*, 50(16):3002–3023, June 2012.
- [13] R. Weber. Mobile gaming website modojo sees record traffic, 2013. <http://www.gamesindustry.biz/articles/2013-03-15-mobile-gaming-website-modojo-sees-record-traffic>.