Poster: AR Game Traffic Characterization – A Case of Pokémon Go in a Flash Crowd Event

Hsin-Yuan Chen
Ruey-Tzer Hsu
National Taiwan University
Taipei, Taiwan

Ying-Chiao Chen
Wei-Chen Hsu
National Taiwan University
Taipei, Taiwan

Polly Huang
National Taiwan University
Taipei, Taiwan
pollyhuang@ntu.edu.tw

ABSTRACT

Latency is a major issue towards practical use of augmented reality (AR) in mobile apps such as navigation and gaming. A string of work has appeared recently, proposing to offload a part of the AR-related processing pipeline to the edge [8]. One pitfall in these studies is the (simplified) assumption about the network delay. As a reality check and to gather insights to realize AR in real time, we seek in this work a better understanding of how a popular AR game, *Pokémon Go*, delivers its data in situ.

CCS CONCEPTS

• **Networks** → **Network measurement**; • **Human-centered computing** → **Empirical studies in ubiquitous and mobile computing**.

KEYWORDS

Traffic Characterization, AR Game, Edge Offloading

ACM Reference Format:
Hsin-Yuan Chen, Ruey-Tzer Hsu, Ying-Chiao Chen, Wei-Chen Hsu, and Polly Huang. 2021. Poster: AR Game Traffic Characterization – A Case of Pokémon Go in a Flash Crowd Event. In *The 19th Annual International Conference on Mobile Systems, Applications, and Services (MobiSys '21), June 24–July 2, 2021, Virtual, WI, USA.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3458864.3466914

1 INTRODUCTION

Pokémon Go is the most popular AR game[1]. The game spawns virtual pocket monsters (Pokémon) and adds game-specific landmarks on top of the physical world. The players, in order to discover the Pokémons or to interact with the landmarks, have to go places in the physical world, encouraging outdoor activities as a result. Such a game design and unique gamer experience draws in diverse players spanning gender and age groups. Since its debut in 2016, the app has been downloaded 1 billion times by March 2019. The annual revenue stands the 7th in the global mobile game market in May 2020[2].

[1]https://www.gamedesigning.org/gaming/augmented-reality/
[2]https://www.statista.com/topics/1906/mobile-gaming/dossierSummary

MobiSys '21, June 24–July 2, 2021, Virtual, WI, USA

ACM ISBN 978-1-4503-8443-8/21/06…$15.00
https://doi.org/10.1145/3458864.3466914

Figure 1: Flash Crowd in a Safari Zone Event

The use of AR mode in Pokémon Go is however limited. In AR mode, a player sees the Pokemon rendered in front of the physical environment within the camera view. The only interaction allowed is the player whirling and throwing a ball on the phone screen, trying to capture the Pokemon. The AR+ mode[3] is an early attempt to engage the virtual creatures and the players more using the camera as a range sensor. The sensing error is so large that it fails to support the intended game play – detection of the player approaching a Pokémon. With that said, recent developments of high-precision sensors and ranging mechanisms [3] [4] [6] [5] , game designs along the line of AR+ model will rise, enriching the game play. Rising in the meantime is also the potential of vision-based techniques [1] [2] to enable finer-grained interaction between the game creatures and the physical surroundings. The challenge less addressed is the increased traffic demand and its impact to network latency.

As a reality check and towards AR in real time, we recorded the game traffic from 6 volunteers in a *Safari Zone* event taking place in the Metropolitan Park of New Taipei City in October 3-6, 2019[4]. See **Figure 1**. The visitor count is approximately 890,000 over the 4-day span. The server, access network, and client were in effect undergoing stress tests[5].

2 MEASUREMENT METHODOLOGY

The challenge of collecting packet traces in the wild is the diversity and the level of manipulation required to volunteers' smartphones. For Android phones, rooting is required to install a packet sniffer.

[3]https://www.forbes.com/sites/davidthier/2017/12/21/pokemon-gos-new-ar-on-ios-is-actually-cool-but-needs-work/?sh=290d3f5c4733
[4]https://pokemongolive.com/post/ntc-safari-zone-2019/?hl=en
[5]https://comicbook.com/gaming/news/pokemon-go-fest-slow-lag/

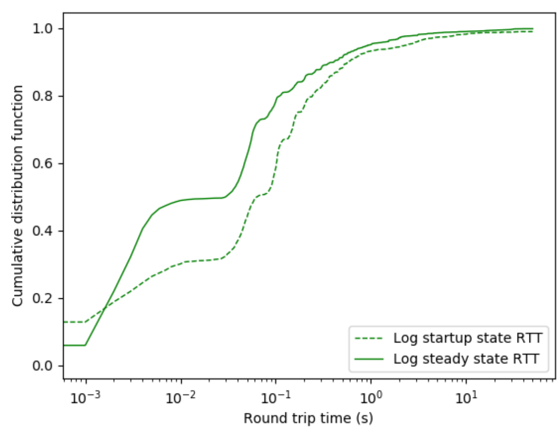

Figure 2: RTT in Startup State vs. Steady State

For iphones, rooting can be avoided, but a MacOS device is required to set up a virtual tunnel to the phone. Having only a 4-day window, we took a platform-independent and manipulation-free approach to maximize the recruit rate. In that, the volunteers' smartphones are connected through WiFi to an LTE hub. The hub relayed the traffic between the mobile client to the game server. The measurement was taken on a laptop connected also to the WiFi subnet. A tcpdump [6] process ran on the laptop and sniffed all packets passing through the subnet. The laptop was carried by a researcher who followed the players closely throughout the experiment. Six players agreed to participate and played as a group for an hour.

159,966 packets were recorded. 68,554 of them were Pokémon Go traffic. To separate the traffic from app to app, we surveyed the server IP addresses observed in the trace and compared them against the server IP addresses used by well-known apps. Pokémon Go traffic was particularly easy to single out as the game used a permanent, anycast IP address[7].

3 PRELIMINARY ANALYSIS

In this preliminary study, we examine two traffic metrics that are relevant to how the AR module decomposes its computation task and transmits the necessary data for low latency. They are the bandwidth consumption and RTT.

Bandwidth Consumption. The level of bandwidth consumption is distinctive when a player is in the startup vs. the steady state. For players who are new to the playground, information such as the area map and cached user data need to be updated before the game session can begin. This startup traffic is significantly higher in volume. The average uplink and downlink bandwidth are 3.27 and 7.57kbps per player. The peak downlink demand can go up to 17kbps. In the steady state, they are only 0.6 and 1.48kbps per player, with the peak downlink demand at about 6kbps. The bandwidth requirement is higher than the conventional online games but similar to other mobile games [7].

[6] https://www.tcpdump.org/
[7] https://owenthe.dev/2016/07/blocking-pokemon-go-from-my-network-how-i-did-it-and-how-you-can-too/

Round Trip Time (RTT). Mass majority of the RTTs are short – about 80% of the RTTs are less than 100ms. To most players, it feels the server responds in real time for most part of the event. Furthermore, 44.58% of the RTTs are within 10ms. This suggests that the server is well situated and very close to the clients. When the traffic load is light, the network delay is indeed negligible.

The long RTTs are extremely long however. This is very likely the result of network saturation. In that, buffering and re-transmission delay are substantially long and dominate the end-to-end delay. Here we separate the startup and steady state traffic. **Figure 2** shows the distributions of RTT in startup vs. steady state. Note the log scale on the X axis. 29.67% of the RTTs are within 10ms in the startup traffic and 48.49% in the steady state traffic. The volume of the startup traffic does play a role prolonging RTTs.

The volume of the startup traffic, however, plays a much lighter role stretching the RTTs that are already long. We see in the data that the difference in the proportion of RTTs within 10ms between the startup and steady state traffic is 18.82% and that for RTTs within 200ms is only 7.983%. The former being large suggests that the volume of the startup traffic does add to the delay. The latter being much smaller indicates that the startup traffic, despite the high volume, is not the main cause of the extremely long RTTs. *Traffic concentration* might be the true devil impairing the gaming experience. Throughout the measurement, the volunteers play in a group nearby the data logger. Their actions towards the game spawns and landmarks appeared similar and synchronized.

4 FINDINGS AND OUTLOOK

Two major findings are: (1) Network delay varies significantly. A offloading mechanism might need to adapt accordingly to meet the end-to-end latency budget. (2) Traffic volume, as well as traffic concentration, impact network delay. Additional compression and randomization schemes might help mitigating the long RTT problem. Moving on, our plan is to expand the data collection and analysis effort, for more insights to AR app design and a traffic or user model to realistic evaluation of the AR apps.

REFERENCES

[1] Huidong Bai, Gun A. Lee, Mukundan Ramakrishnan, and Mark Billinghurst. 2014. 3D Gesture Interaction for Handheld Augmented Reality. In *SIGGRAPH Asia 2014 Mobile Graphics and Interactive Applications (SA '14)*. Article 7.
[2] Hayet Belghit, Abdelkader Bellarbi, Nadia Zenati, Samir Benbelkacem, and Samir Otmane. 2015. Vision-Based Collaborative & Mobile Augmented Reality *(VRIC '15)*. Article 23.
[3] Jason Gao and Li-Shiuan Peh. 2016. A smartphone-based laser distance sensor for outdoor environments *(ICRA '16)*. 2922–2929.
[4] Antonio Jiménez and Fernando Seco. 2016. Comparing Decawave and Bespoon UWB location systems: Indoor/outdoor performance analysis *(IPIN '16)*. 1–8.
[5] Ioannis Pefkianakis and Kyu-Han Kim. 2018. Accurate 3D Localization for 60 GHz Networks. In *Proceedings of the 16th ACM Conference on Embedded Networked Sensor Systems (SenSys '18)*. 120–131.
[6] Navid Tadayon, Muhammed Rahman, Shuo Han, Shahrokh Valaee, and Wei Yu. 2019. Decimeter Ranging With Channel State Information. *IEEE Transactions on Wireless Communications* (05 2019).
[7] Hu Yan. 2016. *Research Report on Pokémon Go's Requirements for Mobile Bearer Networks*. Huawei.
[8] Shuochao Yao, Jinyang Li, Dongxin Liu, Tianshi Wang, Shengzhong Liu, Huajie Shao, and Tarek Abdelzaher. 2020. Deep Compressive Offloading: Speeding up Neural Network Inference by Trading Edge Computation for Network Latency *(SenSys '20)*. 476–488.