5-2015

Matchmaking Game Players on Public Transport

Nairan ZHANG
University of Wisconsin-Madison

Youngki LEE
Singapore Management University, YOUNGKILEE@smu.edu.sg

Rajesh Krishna BALAN
Singapore Management University, rajesh@smu.edu.sg

DOI: https://doi.org/10.1145/2751496.2751503

Matchmaking Game Players On Public Transport

Nairan Zhang‡*, Youngki Lee†, and Rajesh Krishna Balan†
‡University of Wisconsin-Madison, †Singapore Management University

ABSTRACT

This paper extends our recent work, called *GameOn*, which presented a system for allowing public transport commuters to engage in multiplayer games with fellow commuters traveling on the same bus or train. An important challenge for *GameOn* is to group players with reliable connections into the same game. In this case, the meaning of *reliability* has two dimensions. First, the network connectivity (TCP, UDP etc.) should be robust. Second, the players should be collocated with each other for a sufficiently long duration so that a game session will not be terminated by players leaving the public transport modality such as bus or train. In the *GameOn* paper, we focused on the first dimension while in this paper we describe detailed constraints and policies used by the matchmaker and discuss the influences of using *GameOn* on game design.

Categories and Subject Descriptors

C.2.3 [**Computer-Communication Networks**]: Network Operations — *Network management*; K.8.0 [**Personal Computing**]: General — *Games*

General Terms

Design, Experimentation, Measurement

Keywords

Matchmaking; p2p Games; Mobile Gaming; Public Transportation

1. INTRODUCTION

Playing mobile games becomes a major leisure recreation. With the development of the mobile device industry and the prosperity of game distribution platforms (e.g., the Google Play and the Apple Store), we believe the trend will continue.

Multiplayer game, in particular, is a very popular form of mobile games. For example, the Entertainment Software Association stated, in 2014 [1], that 62% of U.S. American gamers played games with other people. Advantages of multiplayer games over singleplayer games are twofold. First, it brings more funs into game plays thanks to the competitive battles with real people. Second, it introduces the social awareness into playing activities. Correspondingly, there are two matchmaking challenges when creating a compelling multiplayer game experience: 1) matching players with compatible and complementary skills and talents (i.e., people want to play with their friends and with people as skilled as they are), and 2) matching players with similar and adequate network latencies.

Both these challenges are already active lines of research. For example, Switchboard [6] proposed techniques to predict latencies so that players with similar latencies could be grouped together. Xu *et al* [7] presented the understanding on the importance of social relations between game players.

We recently proposed the *GameOn* [8] system that tried to boost both at the same time. The rational behind *GameOn* was that, in many urban cities, commuters spend a large part of their day on a train or a bus. Thus, it would be great if we availed them of the opportunity to engage in spontaneous multiplayer gaming with their fellow commuters on the same train or bus, and thus potentially meet people with shared interests.

To start a game, *GameOn* uses a cloud-assisted matchmaking phase where players send requests to a central matchmaker via cellular connections. The matchmaker collects various information (e.g. peer link qualities, peer collocation time) from players and uses that to form reliable game groups for the players. Here, we use two different definitions for *reliability*; first, the network connections between all game players must be robust (i.e. have sufficient bandwidth and low enough latency), and second, the group players should be collocated with each other for a sufficient long duration so that a game session will not be terminated by players leaving (i.e. they exit from the train or bus). Once a game is started, *GameOn* uses p2p protocols (either Bluetooth or Wi-Fi Direct) to connect all the players together — thus eliminating the long latency potentially expensive cellular links while providing free low latency communications.

In [8], we showed that *GameOn* can network players with reliable p2p connections. However, a question we did not answer is how to improve the second dimension of *reliability*, i.e., forming a group with sufficient long collocation time. In this paper, we will show how *GameOn* can use transport usage data, along with additional inputs collected in the p2p environment, to build an effective matchmaker. We present grouping policies and customizable matchmaker to address these challenges. Next, we discuss some new challenges (conflicts caused by our policies) as well as opportunities (new types of game modes) that arise due to the proposed matchmaker.

*This work was done when he was a Ph.D. intern at Singapore Management University.

MobiGames'15, May 18, 2015, Florence, Italy.

ACM 978-1-4503-3499-0/15/05 ...$15.00.
http://dx.doi.org/10.1145/2751496.2751503.

2. BACKGROUND AND RELATED WORK

Matchmaking is a key component in p2p multiplayer games. It is a service that discovers and groups proper players together so that game plays can be more enjoyable. The matchmaker takes different constraints into account and finally distribute a grouping decision to players. In this section, we first review constraints that a typical matchmaker considers and then describe new aspects introduced by *GameOn* .

There have been many research works and commercial platforms that explored different inputs for a matchmaker to make proper grouping decisions. Overall, we categorize them into following three bins.

- **Player-specific inputs.** Before a game play starts, players have individual expectations on the opponents. For example, a player from Japan may want to play games only with other Japanese people for the sake of a communicable language. Other examples include demographic similarity, geographic closeness, friends, frequently-met peers and so on. Multiplayer games should provide well user interface for players to specify their needs, and importantly, such needs are usually satisfied in the first place.

- **Network-specific inputs.** A matchmaking service is built for a specific network environment. That is, deploying a game on different network medias and / or topologies poses distinct challenges for implementing a matchmaking service. For example, since the 3G/LTE network performance is greatly variable [5], achieving short and predictable communication latencies between a player and the game server becomes unprecedentedly important. A p2p matchmaker thus needs to carefully elect the game server and group peers according to the network latency. Switchboard [6] presents a scheme to estimate latencies rather than frequent measurements. that will incur a lot of cellular network traffics.

- **Game-specific inputs.** To increase fun and user stickiness, a multiplayer game may have specific requirements when matchmaking players. A basic one is grouping players according to their levels of skills. This is fairly important for playing battle games. By analyzing the history performance of each player, a ranking system running on a centralized server can estimate current skill levels of players. As an example, Xbox Live [2] employs the TrueSkill algorithm [4] that identifies the skills of gamers.

3. MATCHMAKING CONSTRAINTS

Working for in-transport network environment, a *GameOn* matchmaker encounters a set of new challenges that are not concerned in prior matchmaking work. We begin by introducing the use cases of *GameOn* .

3.1 GameOn Use Cases

GameOn is a multiplayer matchmaking service that helps discover co-players within the same train or bus. It enables commuters to send game play requests to the matchmaker and groups them in a reasonable way. At the same time, *GameOn* elects a host from the group members, which, besides participating in the common game play, maintains and distributes the group status over the entire game session. *GameOn* uses p2p network techniques such as Wi-Fi Direct and Bluetooth to avoid long latencies of the cellular network. Therefore, *GameOn* has to take the peer connectivity into consideration so that peers can reach out to each other directly once a game session gets started. Although such a matchmaking protocol works outside the public transport scenario, considering the maximum range of a p2p connection, *GameOn* focuses the discussion on the in-train or in-bus scenarios.

Another challenge is the *GameOn* matchmaking service is built on an ephemeral network in which both links and nodes emerge randomly and do not exist permanently. The matchmaking goal for such a setting is achieving high group *reliability* that is not seriously exposed in preceding matchmaking works. In this case, the reliability has two dimensions: 1) the p2p network connectivity (TCP, UDP etc.) should be robust, and 2) the players should be collocated with each other for a sufficiently long duration so that a game session will not be terminated by players leaving the public transport modality (train, bus, etc.). In [8], we answered the first half of the question, that is, how to discover and set up robust p2p connections, while in this paper we will address the second half.

3.2 Emerged Matchmaking Constraints

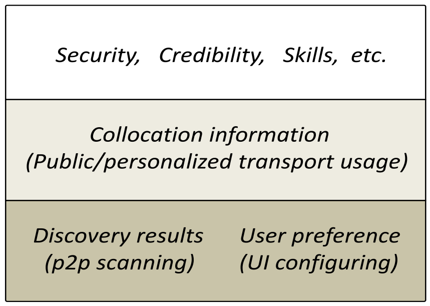
The public transport setting in question poses several *pre-conditions* for a *GameOn* matchmaker to satisfy.

1. **Players may specify more detailed needs such as gender dissimilarity and working place closeness.** To integrate more social funs, a matchmaker can honor matches based on fine-grained demographic and geographic information. This can be achieved by collecting more personal information, which raises security-related issues.

2. **A player has to able to discover collocated peers.** This is a necessary condition for any player to be connected into a multiplayer game. Since the discovery process is accomplished before collocated peers are truly connected, the discovering result implies the network environment within the modality at time. By interpreting this result, the matchmaker can tell which connections are *probably* easier to build and robuster. *GameOn* uses the discovery time as the indicator, that is, the shorter discovery time means a stronger connection. However, if the connection was not successfully built, the player has to ask the matchmaker for a new grouping decision.

3. **Client players should be connected reliably to the server player.** During a game play, connections within a group can experience various interferences. A dominant interference results from alighting and boarding of irrelevant passengers. To make a game session not terminated by such interferences, the matchmaker should be able to approximately guess players' locations and distances in-between so that connections are short with good quality. Moreover, the matchmaker can adapt the grouping decisions to various locations where alighting and boarding can be predicted.

4. **Players should keep connected with each other for a sufficient long duration.** There are various possible reasons to explain a disconnection. Some of these causes are client-specific such as battery concerns and escape cheating. *GameOn* can use heuristics to understand the real reason. On the other hand, there is also another compelling reason for disconnections that are caused by player alighting. It is possible for the matchmaker to figure out the alighting station of a particular passenger based on his or her history usage. However, we did not present concrete grouping policies in [8]. In this paper, we attempt to answer if disconnections caused by alighting behaviors can be avoided by analyzing history data and strategically grouping passenger players.



Traditional constraints now become secondary and should be satisfied after group reliability is guaranteed.

Figure 1: A Layered Matchmaker Used By *GameOn* .

In summary, to meet aforementioned pre-conditions, *GameOn* adds three ingredients for a game developer to design a *GameOn* matchmaker, namely, more specific user preferences, robust peer connections and predicted collocation times.

4. GameOn MATCHMAKER

Matchmaking players for online games is not a new topic. Mathematically, the matchmaking process is an optimization problem with a specific objective. For example, since the network performance affects game experiences, a well-recognized goal is to group players in such a way that the network latency between players should be minimized. In *GameOn* , however, since some constraints ride on others, the importance of individual constraints are not equivalent.

4.1 A Layered Matchmaker

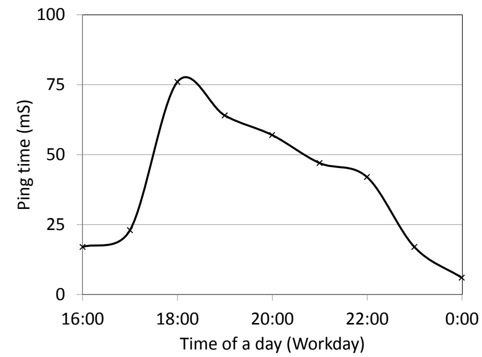
Figure 1 shows the layered matchmaker inputs. The first and foremost is the bottom layer that includes peer discovery and user preference information. Peer discovery information is required because *GameOn* uses p2p network techniques, while user preference represents user expectations when using *GameOn* . Once peer connections are successfully built, the middle layer is used to predict the group coherence, i.e., how long a group can exit. As we will show in the following subsection, *GameOn* utilizes transportation usage data to predict the collocation time. When public or personalized transport usage data are available, different levels of grouping polices can be applied correspondingly. The top layer includes other inputs from the traditional criteria set. These matchmaking needs can be achieved only when the lower layer requirements are already satisfied.

Game developers can configure matchmaking policies through *GameOn* platform. *GameOn* provides various classifiers such as logistic regression and support vector machine. Developers can call these algorithm modules. The history data is stored in *GameOn* storage infrastructure.

In the *GameOn* paper, we have already presented our solutions to discovering and building reliable connections. Also at the bottom layer, meeting user preferences is straightforward. In this paper, we focus our discussion on how to utilize transport usage data to predict group coherence.

4.2 Grouping Policies

The group development in a transit modality can be highly dynamic since players have high variances on their transport usages. This introduces new challenges to grouping multiple players. One important concern before designing an effective matchmaker is the boarding and alighting behaviors may happen too frequent so it is less likely for an already-formed group to last sufficient long time.



We measured the ping time during a short trip including 3 consecutive stations along the yellow line. In particular, they were *Esplanade*, *Promenade*, and *Nicoll Highway*. The ping time indicates the level of in-train crowdedness. Visually, it touches the highest point around 6pm at a workday and gradually reduces with time.

Figure 2: The Human Traffic In and Out A Station (i.e., Board and Alight A Train) Has Clear Temporal Pattern.

If this is the case, then game experiences in such setting will not be satisfying. The *GameOn* matchmaker must carefully select group members so that game plays are less likely to be terminated due to player alighting.

Unlike client-specific reasons for a disconnection, disconnections caused by alighting are more predictable if we reasonably use transport usage data. By doing so, alighting disconnections can be avoided by strategically grouping passenger players and electing a host player. It should be noted that some disconnections do not affect the game experience of other people in the group. For example, in a shooting game client disconnections actually happen quite often in real world. The host and other clients do not worry about this, provided that their scores can be recorded correctly.

We now present two example policies that are used for grouping players. The first policy uses only public transport usage data such as peak hours and busy stations while the second policy assumes personalized transport usage data can be available, and uses them to make a matchmaking decision.

4.2.1 Alighting-aware Policy Using Public Transport Usage Data

For daily commuters, a primary usage of the public transport is commuting between home and work. Since a transportation system in a city is built on the city planning, the patterns of traffic flows can be easily recognized. For example, the central area as the financial center undergoes high traffic at peak hours. Similarly, it is also not difficult to recognize traffic patterns with time at shopping malls and schools. Clearly, each region has its identity (e.g. business, recreation, schools) in the city, and we hypothesize that traffic flows in and out these regions are highly predictable.

We conducted a real-world experiment to validate this. We chose yellow line trains in Singapore, which consisted of 3 carriages. Each carriage was filled with numerous metallic objects (seats, hand rails, guard rails etc.) and was 23 meters in length, 3.2 meters in width, and 2.1 meters in height with a very small (negligible) inter-carriage gap. We then connected a Galaxy S3 (running Android 4.3) with a S5 smartphone (running Android 4.4.2) via Wi-Fi Direct connection to measure the inter-phone ping time on the trains. Two devices were two carriages (46 meters) away from each other.

We picked three consecutive stations to conduct measurements. The middle one was an interchange station. We measured the ping time every hour (on different trains) from 4pm to midnight. The goal was to understand, *for a particular station*, how the human traffic (equivalent to in-train crowdedness) varies with time. Figure 2 shows the temporal pattern is fairly obvious. This motivates us to use city planning information.

Based on these common available data, we can capture important measures of public transportation usage such as how many passengers newly getting on and off at stations at a particular time, so called *public transport usages*. Then the matchmaker can decide how long a group can exist, and when and where more players will probably join the *GameOn* platform.

In particular, consider each passenger i is taking route l at time t. Suppose that for this particular line and time, the public transport usage is denoted by $T^{t,l}$ that can be drawn from public transport history data. Then we assume that the passenger's alighting station $A_i^{t,l}$ can be determined by the possibility:

$$p(A_i^{t,l}|T^{t,l}) = \mathcal{N}(T^{t,l}, \sigma^2). \quad (1)$$

4.2.2 *Alighting-aware Policy Using Personalized Transport Usage data*

The public patterns are generic, and cannot represent personalized alighting/boarding behaviors. Integrating personal data into the decision making process can improve the prediction accuracy. For example, we can precisely predict which station one alights, or how long a particular client will remain on the train / bus. These values can be computed using historical data (using techniques similar to Balan *et al* [3]).

In particular, suppose each player has an unknown transport usage $S_i^{t,l}$ Then we assume that the passenger's alighting station $A_i^{t,l}$ can be determined by the possibility:

$$p(A_i^{t,l}|S_i^{t,l}) = \mathcal{N}(A_i^{t,l}; S_i^{t,l}, \sigma^2). \quad (2)$$

There are also other ways to get the alighting station. For example, an alternative way is asking players to specify the alighting station when they log in the *GameOn* platform. Or when a user requests to play a certain game, *GameOn* shows the typical length of this game as a warning and lets the player decide to join or wait. However, we do not assume players are cooperative.

4.3 Simulation Results

We built a simulator to quantitatively understand grouping and host election.

The first simulation was to compare different policies. Besides above two policies, we will also show the results of not using any policy — only p2p discovery information can be used. The simulation started from one end of a transport line, and ends at the other end. Each group had a size four and each game session lasted two minutes. Over the line we had 20 stations in total and the station interval was 2 minutes. Suppose that players joining the *GameOn* platform follows a Poisson distribution. Further suppose that the time length a player staying on the platform follows a normal distribution with the mean station to alight at the CBD place.

Grouping policy 1 is not using transport usage data. Policy 2 allows *GameOn* to know on which station is the CBD. Finally, policy 3 allows *GameOn* to know exactly which station a particular player will get off the train.

It is intuitive to define the group reliability r as the percentage of disconnected game plays D_S in all game plays S. Obviously, to maximize the group reliability, one should strive to reduce D_S. Figure 3 shows the r values of three policies.

$$r = 1 - \frac{D_S}{S} * \% \quad (3)$$

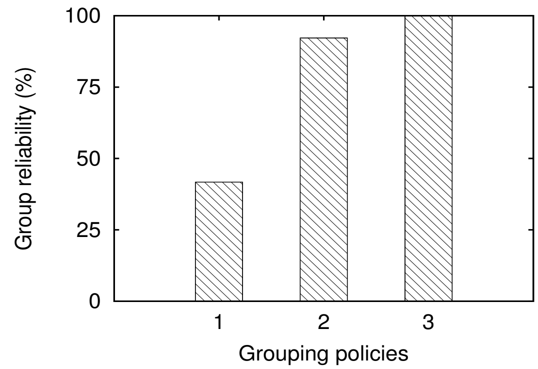


Figure 3: The Group Reliability of Three Policies.

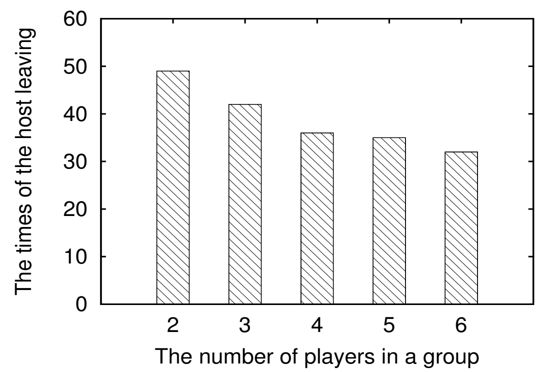


Figure 4: The Times of Host Leaving Reduces As The Group Size Increases.

As mentioned earlier, in some games only host leaving terminates the game. The next simulation was to check the relation between the group size and the times of host leaving. For each run of the simulation, the matchmaker grouped players with a different group size from 2 to 6. The matchmaker also randomly elected a player as the game host. We counted the times of the host leaving that leads to a group completely broken. Note that in this simulation we assume a client leaving does not lead to a group broken. Figure 4 shows that when large groups were formed, the times of host leaving was reduced. This means a game session is more likely gracefully finished. This is primarily because the required number of hosts decreases due to a larger group size. Consider bots can be allowed to replace a client as well as well-designed game session length, groups formed by *GameOn* can be fairly stable over a game session.

4.4 Discussion

This paper and [8] only focus on the technologies to enable *GameOn* . However, designing a real *GameOn* matchmaker will face more serious problems. For example, various matchmaking objectives are usually conflicting. This level of considerations has not been addressed. The first conflict is between security and sociality. Although an original motivation of building *GameOn* is to increase social awareness, security cares must be taken since group members are physically collocated (consider one may pervert kids). Thus, a player may not like to be grouped with other players with very close proximity. Similarly, only focusing on potential security issues without any social fun may degrade the value of *GameOn* .

Next, when *GameOn* allows fine-grained user preferences on security and sociality, e.g., *group me only with students in my school*, these two objectives will definitely conflict with the availability. Clearly, there is little chance for players to be grouped together with too many constraints.

All these issues were raised when we designed a *GameOn* matchmaker. In the future work, we are very interested in investigating more social aspects associated with a system like *GameOn* .

5. INFLUENCES ON GAME DESIGN

In previous sections, we discuss how a *GameOn* matchmaker will group collocated players. Another key question is how game design can be affected because of using *GameOn* . In this section, we describe the influences caused by *GameOn* and share our experience of using *GameOn* .

- **The length of game sessions.** It is well-known that casual games mostly have level-driven game modes. Starting from a lower level, one can challenge more difficult levels after he successfully solves the lower levels. However, if the time required to finish one single level was too long, then a game session may not be completed before any player gets off.

  It is clear that if the game session is short, say 1 or 2 minutes, the risk of group broken becomes lower. This challenge requires developers to carefully design the length of a game session. Our experience is a time-driven game mode may be more suitable for the transport settings, in which the winner will be the one who gains the highest score within a given time limit (e.g. two minutes).

- **The start and end of a game session.** In [8], we found existing connections were greatly interfered when at stations because irrelevant passengers board and alight. Therefore, only shortening the length of game sessions are not enough.

  If taking this into consideration, a game developer may forcibly start game sessions right after trains' embarking, and end them right before the next stop. Loser(s) of the current game session will be put into a waiting queue at the next round if they are still on the train.

- **Group size.** Another issue is real time battle games like *Quake 3* are more appealing when the group size becomes larger. However, as we described in [8], the collocation time reduces when the group size increases. Moreover, since all game logics (both the server logic and client code) runs on resource-constrained mobile devices, it is difficult to support the large groups with size greater than 6.

  However, we also found, for the games in which bots are allowed and client leavings are acceptable (these two conditions hold true in many shooter games), choosing a proper host is more important. Our experience shows that it is better to rotate the host considering restricted mobile resources. This requires more supports from the network layer.

- **User interface.** We first considered casual games such as *2048*, *Tetrics* and *Candy Crush* a killer-app for *GameOn* . However, as we extended original *2048* to a multi-player version, we found it was less exciting to see two (not mention to be more) game UI instances on one screen. Especially, considering a selling point of this kind of games is the dazzling animations and graphics, changing UI appearances may be devoid of the point of these games.

  At the developer side, having multiple players (or their virtual representatives) on one screen may need redesigning the game UI code. A design principle when integrating *GameOn* with existing games was decoupling our extension library from the game logic, so that developers can easily interface with *GameOn* . This is not difficult for core logics as long as all statuses are well-packaged. However, developers have to re-design the UI code and figure out a reasonable way to interact with users.

- **Ranking systems with more fine-grained types of participants.** In contrast to casual games in which players normally have similar levels of skill, some games such as real-time battles may need grouping players according to the skill level. This is a focus of many preceding matchmaking works. As a general platform, *GameOn* should take the level of skill into consideration for the ranked games. This may further require high user stickiness to the system and thus *big data* used in the data analysis phase can be available.

  The upside building ranking systems is increasing new social ties for commuter gamers. For example, based on *GameOn* we can build a local ranking system for a particular group of commuters (e.g., people riding Route 14 between 8AM and 9AM on weekdays). With continuously collected user skill values we are able to match them in this sense.

6. CONCLUSION

The growing trend is that commuters today use smartphones to play mobile games on public transports. We built a system called *GameOn* to ease peer discovery so that playing multiplayer games on a train or bus is possible. In our earlier work [8], we addressed the issue of find and setup reliable network-layer connections between players, while in this paper, we answered how to find and setup reliable user-layer connections, i.e., maximize the degree of group coherence.

However, when a game running with *GameOn* , it has to adapt to new requirements proposed by the new setting. In this paper, we also discussed *GameOn's* influences on game design, and enabled new game modes.

When building *GameOn* , we use existing games as benchmarks and examples. However, we contend *GameOn* can provide more funs by creating more social-aware mobile games, and thus play as a gateway to the urban gaming. For example, riding on a traditional game play mode called *Capture The Flag* that typically has transportation tools included in the game, *GameOn* can blend virtual and physical worlds, and enable passengers to virtually capture the flag in the train or somewhere in the city. *GameOn* can also be an entry point to games in the physical world such as passengers can complete special missions in the morning. As this scenarios come true, we believe that the necessity of *GameOn* will become crucial.

7. ACKNOWLEDGMENT

We sincerely thank anonymous reviewers for their valuable feedback. We also thank Lixuan Zhai, Xiaonan Guo, Nguyen Minh and Shriguru Nayak for their helps to perform the in-train measurements. This work was supported partially by the Singapore Ministry of Education's Academic Research Fund Tier 2 under research grant MOE2011-T2-1001, and partially by the National Research Foundation, Prime Minister's Office, Singapore, under the IDM Futures Funding Initiative. Any opinions, findings, conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the granting agencies or Singapore Management University.

8. REFERENCES

[1] `http://www.theesa.com`.

[2] `http://www.xbox.com/en-US/live`.

[3] R. K. Balan, K. X. Nguyen, and L. Jiang. Real-time trip information service for a large taxi fleet. In *Proceedings of the 9th International Conference on Mobile Systems, Applications, and Services (MobiSys)*, 2011.

[4] R. Herbrich, T. Minka, and T. Graepel. Trueskill(tm): A bayesian skill rating system. In *Advances in Neural Information Processing Systems 20 (NIPS)*, 2007.

[5] K. Lee, D. Chu, E. Cuervo, Y. Degtyarev, S. Grizan, J. Kopf, A. Wolman, and J. Flinn. Outatime: Using speculation to enable low-latency continuous interaction for mobile cloud gaming. In *Proceedings of the 13th International Conference on Mobile Systems, Applications, and Services (MobiSys)*, 2015.

[6] J. Manweiler, S. Agarwal, M. Zhang, R. R. Choudhury, and P. Bahl. Switchboard: a matchmaking system for multiplayer mobile games. In *Proceedings of the 9th International Conference on Mobile Systems, Applications, and Services (MobiSys)*, 2011.

[7] Y. Xu, X. Cao, A. Sellen, R. Herbrich, and T. Graepel. Sociable killers: understanding social relationships in an online first-person shooter game. In *Proceedings of the ACM 2011 conference on Computer supported cooperative work (CSCW)*, 2011.

[8] N. Zhang, Y. Lee, M. Radhakrishnan, and R. K. Balan. Gameon: P2p gaming on public transport. In *Proceedings of the 13th International Conference on Mobile Systems, Applications, and Services (MobiSys)*, 2015.