A Smart Map Sharing and Preloading Scheme for Mobile Cloud Gaming in D2D Networks

Ziqiao Lin, Zehua Wang, Wei Cai and Victor C.M. Leung
Department of Electrical and Computer Engineering
The University of British Columbia
Vancouver, BC, Canada V6T 1Z4
Email: {t2v0b, zwang, wcai, vleung}@ece.ubc.ca

***Abstract*—With the high popularizing rate of smart devices, mobile gaming is an emerging arena in game industry with the purpose of providing ubiquitous game services to mobile players. Different from the traditional player who play games on their personal computers with wired local area network or WiFi access, more and more mobile players nowadays prefer to play online games on their smart devices that communicate to cloud servers via wireless cellular networks. Therefore, in the mobile cloud gaming context, the monetary cost of downloading or updating map files in games via cellular networks is a new issue that may effect players' experience. On the other hand, an unpredictable latency may be introduced by the wireless links in the cellular network. In fact, a mobile player can preload the maps that he has high probability to go. Moreover, mobile players nearby can also form a device-to-device (D2D) communication network to share their cached maps. In this paper, we consider the problem that with the limited storage space available on each player's device, how to select the maps on either neighboring devices or cloud server to preload so that the utility of the player can be maximized. We first formula an optimization problem and then present our solution. Simulation results show that our proposed map sharing and preloading scheme can significantly increase the utility received by mobile players.**

I. Introduction

Nowadays, over 1.2 billion people play games worldwide. Meanwhile, with the popularization of smart devices [?] (*i.e.*, smartphones, tablets), more and more people prefer to play games on their smartphones or tablets. Therefore, mobile technology not only provides a simple means of communication but also offers an easy way for entertainment and recreation. However, different from the traditional players who play games on their personal computers with WiFi or wired network access, smart devices may communicate to the cloud servers via wireless cellular networks to request game contents, such as the maps in the game. This brings us the new challenges in mobile cloud gaming context.

First, the amount of data traffic consumed by mobile devices have been increasing for many years. From the annual report released by Cisco, global mobile data traffic grew 63% in 2016. The demand for mobile data traffic is expected to grow 49 Exabytes per month by 2021 [?]. Although the amount of data traffic consumed by games varies for different games, the amount of data traffic consumed by mobile player is a non-negligible part in the global mobile data traffic usage. Thus, for game service providers, it is necessary to reduce the mobile data traffic used for game contents delivery without reducing players' experience. Second, within the mobile gaming context, the monetary cost for game map downloading via cellular networks now is a new issue that may negatively effect players' overall experience. Specifically, for the game which needs to frequently download or update maps, a mobile player may not want to play it until he has the WiFi access. Hence, enabling a player to download or update maps in a ubiquitous manner without introducing monetary costs is desired. Third, the wireless links in the cellular network may introduce an unpredictable latency, which may severely decrease the experience of playing an online game. Therefore, the above three problems have to be addressed when we develop the mobile clouding game services.

Recently, the device-to-device (D2D) communication is proposed as a new communication paradigm and has drawn significant attentions in the research community. In D2D communication, nearby mobile devices in a close proximity can communication directly to each other to share digital files or relay the data from a cellular base station to the cell-edge users. The D2D network features high data rates, reliable communications, powering saving, and bandwidth efficiency [?]. Due to these advantages, it has been shown that D2D communication can effectively offload the mobile cellular network to D2D networks [?]. In this work, we apply D2D communication networks in mobile cloud game context for map preloading. Map preloading refers to the technology that enables the downloading of map data in prior, which will improve players' gaming experience. In mobile scenario, mobile players have no limit on their data plan may enable map preloading functionality for their games. Under this assumption, we consider the map preloading problem by taking into account the availability of the maps in neighborhood, the probability of the maps that player will go in the next, the map size, the players' downlink data rate, and players' valuation on gaming experience. To the best of our knowledge, we are the first that propose such a comprehensive model and formulate an optimization problem to solve the map preloading problem. The main contributions in this work are summarized as follows:

- We propose a map sharing and preloading scheme for mobile game players in D2D networks to improve each player's utility. In particular, we first define the utility

978-1-5090-6517-2/17/$31.00 ©2017 IEEE

of a mobile player and then formulate an optimization problem to choose a set of maps for preloading via either D2D or wireless cellular networks.
- We propose a state-of-the-art algorithm with low computational complexity for the formulated map-selection problem.
- We conduct simulation to reveal the performance of the proposed algorithm. results show that the utility of a mobile player can be significantly increased by the proposed state-of-the-art algorithm.

The reminder of the paper is as follows. We review related work in Section **??** and then provide the system overview in Section **??**. The problem formulation followed by the proposed reputation-based fairness scheme is described in Section **??**. Experiments on optimizing the overall latency and enabling the partial execution in an offline manner are conducted in Section **??**. Section **??** concludes the paper.

II. Related Work

A. D2D Content Sharing

The authors in [**?**] proposed a cluster-based D2D network in which files are transmitted from the BS to several clusterheads, and then shared inside each cluster through D2D communication. For the work in [**?**], the author maximized the amount of data traffic offloaded from cellular networks to D2D networks by taking into account mobile users' interest. The author in [**?**] presented the process of precaching files to users to enhance the transmission rate, and then the mobile users can share files with nearby users having the same interest. With the work in [**?**], the author presented a preference-aware content dissemination protocol to maximally match users' preference for content objects. The authors in [**?**] first defined the peers' popularity and proposed a Zipf distribution, which is to rank peers by performance and select strong peers to provide efficient file uploading. In [**?**], the author regarded theoretical D2D content analysis from game theoretic perspective. The D2D content sharing is regarded as a trading network and each mobile device user is a selfish player. The author proposed a bargaining strategy for devices to select their optimal trading strategy in a certain period.

B. D2D in Delay Tolerant Networks

The D2D communication network has been extensively studied in the context of delay tolerant networks [**?**], [**?**]. The delay tolerant network is designated to offer user freedom to access the opportunistic resources. However, its latency does not satisfy the requirements for games. In Table **??**, it summarizes the maximum latency that a player can tolerance before his quality of experience (QoE) is degraded. For an avatar-model game with third perspective, such as World of Warcraft, the player's QoE will significantly decrease when the latency is greater than 500 milliseconds. Basing on these criteria, downloading game contents like maps with D2D communication network may not work. However, when a map that a player has high probability to go in the next scene is currently available on the player's neighborhood. The player

TABLE I
The tolerance of latency in online games [**?**]

Model	**Perspective**	**Example Genres**	**Sensitivity**	**Thresholds**
Avatar	First-Person	FPS,Racing	High	100 msec
Avatar	Third-Person	Sports, RPG	Medium	500 msec
Omnipresent	Varies	RTS, Sim	Low	1,000 msec

can preload the map without monetary cost and use the cached map in the next scene without suffering any latency caused by map downloading.

III. System Model

In this section, we will first present the overview of a novel utility-based map sharing system. Then, we describe the basic settings in our proposed system. Afterwards, the utility model will be presented to jointly consider the benefit received by players for his preloaded maps and the monetary cost if he insist to preload some maps from cloud server via cellular networks.

A. System Overview

The proposed system is designed particularly for multiple scene games. For example, a player in a certain proximity can access other players scene maps except the scene map that they are in right now. And can download other scene maps from other neighbor players in the same scene via D2D network. Specifically, two players in a certain proximity can access other players scene maps. This will relieve the burden of server, minimize the download duration, and even save energy consumption and monetary cost.

However, we do not know which map that a player will use for the next scene *in priori*. Therefore, considering the limited caching space on the user's device, the user's valuation of gaming experience, the user's affordability for wireless cellular data traffic, and the availability of maps in the user's neighborhood, it is a non-trivial problem to select a set of maps and preload these maps via either D2D or wireless cellular communication. In this paper, although the map that a user will use in the next scene is not known in advance, we assume that the probability of a map being used in the next scene is available. In fact, we can categorize players into different types based on their personalities in the game, such as aggressive players and conservative players, etc. Then, we would apply the big data analysis to determine the aforementioned probabilities in a statistical manner.

The main goal is to maximize utility from game players. To be more specific, we will propose a mathematical model and define a user's utility by taking into account the following issues: user's downlink data rate, the caching space available on user's device, the availability of maps on user's neighbors, user's valuation of gaming experience, and user's affordability for cellular data traffic.

Fig. **??** illustrates an example of the considered system for map preloading. There are five mobile players, Players

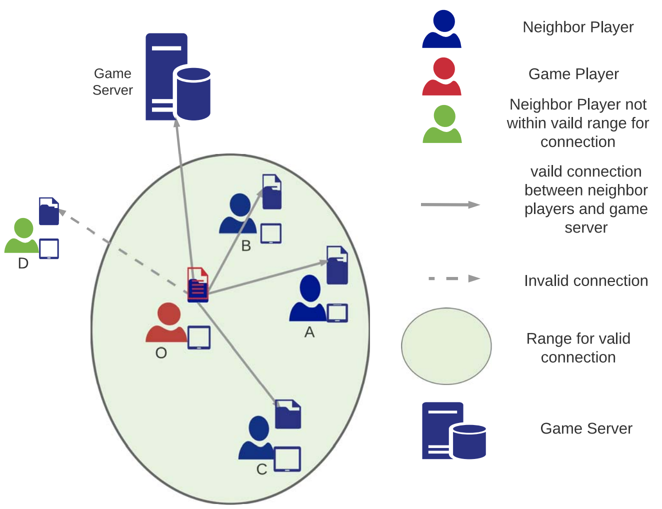

Fig. 1. An Example of Downloading Maps

O, A, B, C, and D, in this example. The circle represents the communication range of Player O located at the centre. Player O can thus access all maps from neighbouring players within his communication range, (i.e., Players A, B, and C). In contrast, Player O cannot access the maps cached by Player D. If a certain map which is not available on neighboring players needs to be preloaded, by Player O can directly download the map from cloud server for preloading purpose.

B. System Settings

We consider a mobile game with multiple scenes and each scene has a map. Let $\mathcal{M} = \{1, \ldots, M\}$ denote the set of all maps in the game. The set of players is denoted by $\mathcal{U} = \{1, \ldots, U\}$. We assume that each player has a smart device and has installed the game. The size of storage reserved for the game is limited on each player's device. Let C_u denote the size of storage that has been reserved for the game on the device of user $u \in \mathcal{U}$. Moreover, we denote S_m as the size of map $m \in \mathcal{M}$. At an instance of time, we assume that each player in set U may have already cached several maps but probably not all of the maps in set $\mathcal{M}$. This is a reasonable assumption due to the limited storage of a smart device. In fact, the practical case is that the maps of a game may be first downloaded and then replaced by other maps to reuse the limited storage reserved for the game. We denote the set of maps which have been cached on the device of player $u \in \mathcal{U}$ by $\mathcal{M}_u$. Thus, for a game with a large number of maps, we typically have $\mathcal{M}_u \subset \mathcal{M}$ and the following inequality holds:

$$\sum_{m \in \mathcal{M}_u} S_m \leq C_u, \quad \forall u \in \mathcal{U}. \tag{1}$$

Besides, we consider that when two players are close enough to each other, they can communicate directly in a D2D manner. In particular, for the current instance of time, we denote $\mathcal{N}_u$ as the set of neighbors of user $u \in \mathcal{U}$. We assume that no monetary cost occurs when a gamer downloads maps from another gamer in his neighborhood.

We now introduce the transitions between different scenes in the game. A player at a given time can just be in one scene and only one map is being used. Sooner or later, the player will go to the next scene, i.e., next map. However, we do not know which map that the play will go *in priori*. In practice, with big data analysis, we can obtain the probability of a map that the player would go to after leaving the current map. We now introduce column vector $\boldsymbol{p}_{u,m} \in \mathbb{R}^{M \times 1}$ to denote the game scene transmission probabilities for player $u \in \mathcal{U}$ who is currently in map $m \in \mathcal{M}$. In particular, the nth ($n \in \mathcal{M}$) element in vector $\boldsymbol{p}_{u,m}$, denoted by $p_{u,m,n}$, represents the probability what user $u \in \mathcal{U}$ goes to map $n \in \mathcal{M}$ after leaving the current map m. Thus, a valid vector $\boldsymbol{p}_{u,m} \in \mathbb{R}^M$ for player u satisfies $\mathbf{1} \cdot \boldsymbol{p}_{u,m}^T = 1$, where $\mathbf{1}$ denotes all-one column vector with a proper size and $\boldsymbol{x}^T$ denote the transpose of vector $\boldsymbol{x}$. Since we consider that the transition happens sooner or later when player u leaves the current map m, we thus have $p_{u,m,m} = 0, \forall m \in \mathcal{M}, u \in \mathcal{U}$.

C. Utility Model

When a game scene transition happens, if the mobile user has already cached the map that he is going to, there is no waiting time for the user to download the new map. On the other hand, if the targeting map is not available on the player's device, the map should be downloaded from the cloud server, where a period of waiting time is usually needed to load the new map. Meanwhile, for mobile players who have only wireless cellular access, the communication link quality in terms of the downlink data rate may fluctuate due the path loss, fading, and interference. That is, an unpredictable latency may be introduced if the map that the player would like to go is not available on his smart device, and the players with poor channel conditions can barely access channel resources since the resources will be allocated to the game players with the best channel conditions [**?**][**?**]. This will decrease the player's gaming experience.

To better server mobile players, the game can preload some maps. As long as the player goes to one of the preloaded maps, better gaming experience can thus be received by the player as the waiting time is reduced. Specifically, if user $u \in \mathcal{U}$ has preloaded map $m \in \mathcal{M}$ and user u does go to map m in the next, the benefit received by user u in terms of the reduced waiting time can thus be defined as $b_{m,u} \triangleq S_m / r_u$, where r_u is the average downlink data rate of user u. It should be noted that the average downlink data rate may vary for different players as their smart devices, wireless service providers, and technologies being used (*e.g.*, 4G, 3G) may be different.

Sometimes, if the map that a player would like to preload is occasionally available in the player's neighborhood, the player can preload the map from his neighboring devices for free. However, if the map is not available in the neighborhood, the player may still want to pre-download the map from cloud server via cellular network. In this situation, a monetary cost may occur. Hence, as long as mobile player $u \in \mathcal{M}$

decides to preload map $m \in \mathcal{M}$ which is not available on his neighbors, the monetary cost can be defined and determined as $c_{u,m} \triangleq S_m q_u$, where q_u is the unit price of wireless data service subscribed by player $u \in \mathcal{U}$. It is worth mentioning that our work actually has taken into account the case that the player has unlimited data plan, where we have $q_u \approx 0$. In summary, the monetary cost can be written in the following form:

$$c_{u,m} = \begin{cases} 0, & \text{if } m \in \bigcup_{v \in \mathcal{N}_u} \mathcal{M}_v, \\ S_m q_u, & \text{if } m \notin \bigcup_{v \in \mathcal{N}_u} \mathcal{M}_v. \end{cases} \tag{2}$$

We take both the monetary cost and the reduced waiting time into account to define the utility of each player that performs the preloading. In particular, if user $u \in \mathcal{U}$ has preloaded map $m \in \mathcal{M}$ and he eventually goes to map m, the utility of user u can be easily determined as $\alpha_u b_{u,m} - \beta_u c_{u,m}$. While, if user $u \in \mathcal{U}$ has preloaded map $m \in \mathcal{M}$ but he does not go to it, the utility is $-\beta_u c_{u,m}$.

IV. Problem Formulation and Proposed Solution

A. Problem Formulation

Since we do not know which map that the user would like to go after leaving the current map, our purpose is to maximize the player's *expected* utility by designing the best preloading strategy. In the end of Section **??**, we have seen that the preloaded map may not contribute to the player's utility unless it is the map that the player eventually goes in the next scene.With out loss of generality, we consider a set of maps $\mathcal{L}_u \in \mathcal{M}$ that user u_x would like to preload besides those maps that user u has already cached. Denote g_u, m as the map that user u is currently staying at, the expected utility of the player is given by

$$\sum_{m \in \mathcal{L}_u} \left(P_{u,g_u,m} \alpha_u b_{u,m} - \beta_u c_{u,m} \right) \tag{3}$$

Note that for user u, (**??**) only depends on the set of preloading maps $\mathcal{L}_u$. For notational simplicity, we introduce the following function for user $u \in \mathcal{U} : f_u \triangleq 2^{\mathcal{M} \setminus \mathcal{M}_u} \mapsto \mathbb{R}$. Now, the value of (3) can be simply written as $f_u(\mathcal{L}_u)$. Our problem for user u can thus be formulated as follows:

$$\begin{aligned} &\underset{\mathcal{L}_u \in 2^{\mathcal{M} \setminus \mathcal{M}_u}}{\operatorname{argmax}} \;\; f_u(\mathcal{L}_u) \\ &\text{subject to} \sum_{m \in \mathcal{L}_u} S_m \le C_u - \sum_{m \in \mathcal{M}_u} S_m \end{aligned} \tag{4}$$

The constraint of problem (4) means that the total size of maps chosen by user u to preload should be less than or equal to the remaining storage size that is available on the smart device of player u.

B. Proposed Solution

It is easy to show that problem (4) is an NP-complete problem. In this paper, we omit the proof due to the limited space. In this section, we come up to a state-of-the-art solution based on the greedy algorithm. We jointly consider the availability of the maps in neighborhood, the game scene transition probabilities, the map size, the players' downlink data rate, players' valuation on gaming experience, and the "willingness to pay" of the user (i.e., the affordability of the user for cellular data traffic) in the algorithm design. Our algorithm is presented in Algorithm 1. In particular, in Algorithm 1,we first sort the maps according to the corresponding elements in vector $\boldsymbol{p}_{u,m}$ in a descending order. Let $i \in \mathcal{M}$ denote the index of the element in the sorted vector of $\boldsymbol{p}_{u,m}$, and define one-to-one mapping $r \triangleq \mathcal{M} \mapsto \mathcal{M}$, where $r(i)$ denotes the index of the corresponding map in set $\mathcal{M}$. Next, we present a lightweight algorithm that let mobile user $\in U$ obtain maps basing on the map set ranked by descending probability. Firstly, we will consider whether downloading this map $r(i)$ will exceed the cache size for user u (C_u) (Line 3). Then we will consider whether this map is available from neighbor players. (Line 4). If the map is available, it will bring a positive utility $U_{u,r(i)}$ and we will pre-load this map (Line 5-7). Otherwise, we must check the utility $U_{u,r(i)}$ brought with this map (Line 8) and then decide whether should pre-load this map (Line 9- 13).

Algorithm 1 Heuristic Utility Maximization Algorithm for Player $u \in \mathcal{U}$

1: Sorting the maps according to the corresponding elements in vector $\boldsymbol{p}_{u,m}$ in a descending order.
2: Set $i = 1$ and start to increase i according to number of entering loop
3: **while** $W_u + S_{r(i)} < C_u$ **do**
4:    **if** map $m \in \mathcal{M}$ **then**
5:       Receive the positive $U_{u,r(i)}$ from this map by the function and preload the map
6:       Increase i by one
7:    **else**
8:       **if** utility $U_{u,r(i)} < 0$ OR $S_{r(i)}$ is greater than certain acceptable value **then**
9:          Give up this map
10:          Increase i by one
11:       **else**
12:          Receive the positive $u_{x,m}$ from this map by the function
13:          Increase i by one
14:       **end if**
15:    **end if**
16: **end while**

V. Simulation

A. Experimental Setup

In this section, we reveal the performance of our proposed system by running simulations. Unless otherwise stated, the parameters adopted for simulation are summarized in Table **??**. Trial 1 follows the algorithm that we design here. Trial 2 bases on the algorithm that downloading maps from descending probability ranked map set but only download the maps that are available from neighbor players. Trial 3 will also download maps by the order of ranked map set, but all the maps will be downloaded from game server. Trial 4 is same way to download maps as Trial 3, but all the maps will be downloaded from game server when the user goes to next scene. Furthermore, the util1 – util4 represet Trial 1 – Trial 4 correspondingly in following figures.

TABLE II
SIMULATION PARAMETERS

Data charge	0.7 cent/MB
Data rate(Bandwidth)	2.4375 MB/s
Area	2500 m^2
Parameter in Rayleigh distribution of map size	σ = 5 MB
Number of game players	20
Cache size average	150 MB
Valid area for connection	15 m as radius

B. Effect on Data Rate

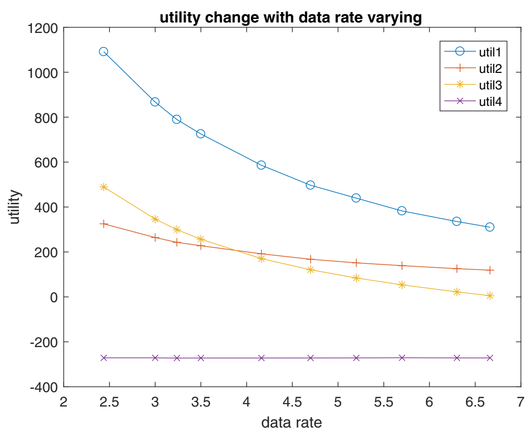

Fig. 2. Utility changes with data rate varying

Since data rate is the denominator for time saving of utility, the positive utility will decrease with increase in data rate. With very high downloading speed, this utility will have less effect and have very small difference to that kind of players that download maps when they go into next scene. However, as we can see from graph, we pre-download maps whatever basing on available map set from neighbor players or map set by descending probability is much better than downloading maps when going into next scene. Trial 2 has an intersection with Trial 3, which means selecting maps basing on probability will have a larger effect than just pre-downloading from available map set from neighbor as data rate increasing. Therefore, Trial 1 will always have a better performance than Trial 3 since they all pre-downloading maps basing on the rank of probability even though pre-downloading has less effect with increasing data rate.

C. Effect on Data Charge

Since data charge is linear relationship to utility, we expect to see all the line should be linear. Trial 2 has no effect on data charge since Trail 2 downloads all maps from neighbor players

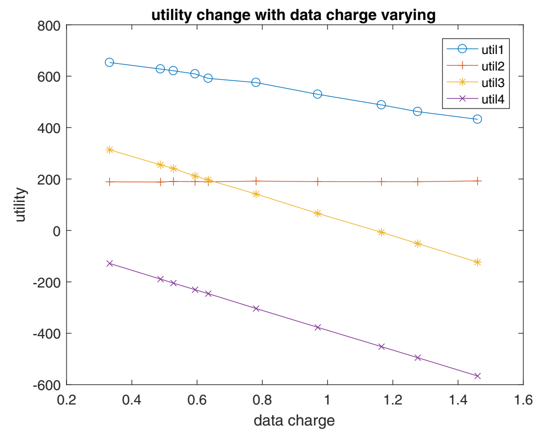

Fig. 3. Utility changes with data charge varying

and there is no cost on data. Since all other 3 algorithms will all depend on data charge completely or partially, these three utilities decrease linearly. In particular, Trail 3 and Trail 4 have a sharp decrasing trend than Trail 1. Since Trial 3 and Trial 4 will download all maps by from server. In this graph, downloading from neighbor has certain effect on the line of Trial 1, and we believe this will have larger effect on Trial 1 if the portion of available maps from neighbor increases.

D. Effect on Area Size

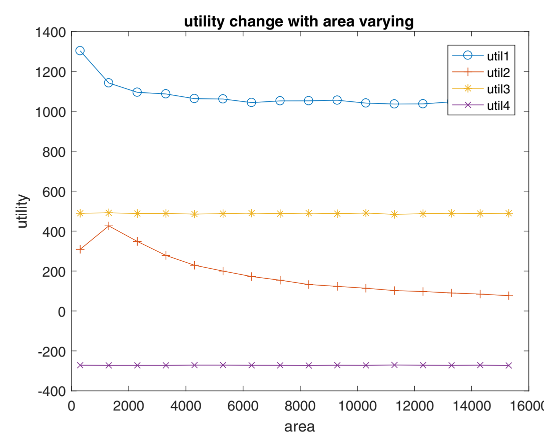

Fig. 4. Utility changes with area varying

We set 20 users and the valid range radius is 15 m as default. As we can see from the graph, as we increase the possible area for users for our algorithm and the utility basing on downloading available map set from other users. The utility wont change for util 2 and util 4 since these two algorithms will download maps directly from game server, which wont get affected by available map set from other game users. Trial 1 and Trial 3 do have a effect by the change of area. However, from the graph, we can see that Trial 1 has smaller effect comparing to Trial 3. And Trial 3 will close to 0 utility since area is too large and available map set may don't have many maps. But Trial 1 still have the highest utility even though it is affected with increasing area.

E. Effect on Cache Size

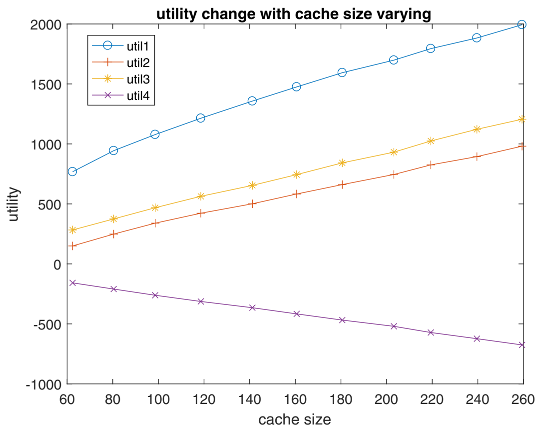

Fig. 5. Utility changes with cache size varying

As shown from Fig. 3, the first three algorithms all tend to increase when users cache size increase. We can see only Trail 4 tends to decrease because user will pay more money and time for the game maps if we can download more maps into our cache. The utility will increase if we download maps either from available map set from other game players or the map set basing on descending probability. Using either one will give the similar trend for utility since Trail 2 has a similar trend as Trail 3. However, trail 1 increases sharply and it shows a sharper trend.

F. Performance of Proposed Algorithm

As we can see from all 4 figures, our proposed algorithm is the best comparing to other way to downloading maps, and there is less negative effect on increment in data charge and data rate comparing to other algorithms, which makes the differences between our algorithm and other algorithm become bigger. Furthermore, our algorithm also has a better performance when cache size increases. When users cache size increase, the utility will also go up nearly linear, and the trend is sharper than other algorithms. In Fig. 4, although our algorithm decreases when the area increases, the decreasing trend tends to be flat and the utility becomes steady in a long run. We may expect to see a cross between our algorithm and others when we continue to increase certain factor. However, the range we covered in this graph is all basing on current game players instrument, and these reach maximum at each factor.

VI. Conclusion

In this paper, we proposed an idea about pre-downloading game maps via D2D communication and we predict the probability of map where game players will go for maximizing the utility for game players. The result shows that our algorithm brings more utility for game players as compared to those without pre-downloading map technique and downloading via D2D. Also, our algorithm shows a less negative effect on data rate,data charge factors and area, and a better performance as compared to others in terms of cache size. However, our work only related to individual player's utility, and try to release data traffic burden from individual perspectives. We see every game player as a selfish player, and help them determine how can they maximize their utility in terms of some factors. However, this in fact may not be the best solution for both game server and all players. Reaching each player's maximum utility does not mean maximizing the server's utility, and maximizing the whole server's utility may become a more complex topic. Data traffic burden may still be a real problem for game server. And we leave this as our future work.

Acknowledgment

This work is supported by funding from the Natural Sciences and Engineering Research Council and UBC Work Learn International Undergraduate Research Award.

References

[1] D. Takahashi, "More than 1.2 billion people are playing games." [Online]. Available: http://venturebeat.com/2013/11/25/more-than-1-2-billion-people-are-playing-games/

[2] C. Inc., "Cisco visual networking index: Global mobile data traffic forecast update, 2016–2021 White paper," Feb. 2017. [Online]. Available: http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/mobile-white-paper-c11-520862.html

[3] K. Doppler, M. Rinne, C. Wijting, C. B. Ribeiro, and K. Hugl, "Device-to-device communication as an underlay to LTE-advanced networks," *IEEE Communications Magazine*, vol. 47, no. 12, pp. 42–49, Dec. 2009.

[4] G. Fodor, "D2D communications ?– What part will it play in 5G?" Jul. 2014. [Online]. Available: https://www.ericsson.com/research-blog/5g/device-device-communications/

[5] D. Wu, L. Zhou, Y. Cai, R. Q. Hu, and Y. Qian, "The role of mobility for D2D communications in LTE-Advanced networks: energy vs. bandwidth efficiency," *IEEE Wireless Communications*, vol. 21, no. 2, pp. 66–71, Apr. 2014.

[6] Z. Wang, H. Shah-Mansouri, and V. Wong, "How to download more data from neighbors? A metric for D2D data offloading opportunity," *IEEE Transactions on Mobile Computing*, to-appear.

[7] N. Golrezaei, A. G. Dimakis, and A. F. Molisch, "Wireless device-to-device communications with distributed caching," in *Proc. of IEEE International Symposium on Information Theory Proceedings*, Jul. 2012, pp. 2781–2785.

[8] K. C.-J. Lin, C.-W. Chen, and C.-F. Chou, "Preference-aware content dissemination in opportunistic mobile social networks," in *Proc. of Proceedings IEEE INFOCOM*, Mar. 2012, pp. 1960–1968.

[9] T. Huang, K. Xu, and R. Pi, "A popularity-based neighbor selection model in P2P file-sharing system," in *Proc. of IEEE International Conference on Broadband Network and Multimedia Technology (IC-BNMT)*, Oct. 2010, pp. 975–979.

[10] Y. Gu, L. X. Cai, M. Pan, L. Song, and Z. Han, "Exploiting the stable fixture matching game for content sharing in D2D-based LTE-V2X communications," in *Proc. of IEEE Global Communications Conference (GLOBECOM)*, Dec. 2016, pp. 1–6.

[11] M. Rouse, "delay-tolerant network." [Online]. Available: http://searchnetworking.techtarget.com/definition/delay-tolerant-network

[12] M. Claypool and K. Claypool, "Latency and player actions in online games," *Commun. ACM*, vol. 49, no. 11, pp. 40–45, Nov. 2006.

[13] X. Ge, H. Jin, and V. C. M. Leung, "Opportunistic downlink scheduling with resource-based fairness and feedback reduction in distributed antenna systems," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 7, pp. 5007–5021, Jul. 2016.

[14] X. Ge, H. Jin, J. Cheng, and V. C. M. Leung, "On fair resource sharing in downlink coordinated multi-point systems," *IEEE Communications Letters*, vol. 20, no. 6, pp. 1235–1238, Jun. 2016.