A Novel Game Map Preloading and Resource Provisioning Scheme in Cooperative Cloud 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 popularization of mobile smart devices (*e.g.,* smartphones, tablets), the number of mobile game players increases significantly. This makes mobile gaming industry rise and take up a large piece of global game market. Different from the traditional players playing games on their personal computers with wired local area network or Wi-Fi access, mobile game players nowadays play online games on their smart devices that communicate to cloud servers via wireless cellular networks. Therefore, the monetary cost for game content downloading and updating via cellular networks may increase the burden of gamers. Meanwhile, an unpredictable latency may be introduced by communication links of wireless cellular networks. This may negatively affect player's gaming experience. In fact, we can predict the next movement of the player in the game and preload the maps that have high probability to go. Meanwhile, if any of these maps have been downloaded and cached by other players nearby, with the Device-to-Device (D2D) communication networks, people may preload these maps freely. In this paper, we focus on the problem that how to select maps to preload from either game server or neighborhood with the consideration of limited storage space on smart devices. We first formulate an optimization problem. Since the formulated problem is NP-hard, we decouple the problem into two subproblems and solve them iteratively to get a suboptimal solution. Simulation results show that our proposed scheme can significantly increase the utility received by mobile players.**

I. INTRODUCTION

With the popularization of smart devices, more and more people choose to play games on their smart devices (*e.g.,* smartphones, tablets) via wireless cellular networks [1]. Different from traditional game players who play games on their personal computers with Wi-Fi and wired network access, smart devices can directly communicate to the cloud servers via wireless cellular networks to request game contents, such as maps in the game. However, mobile games not only bring the opportunities for computer game industry but also find some new challenges need to solve.

Firstly, mobile technology brings a serious data traffic burden. According to the Cisco Visual Networking Index, global mobile data traffic grew 63% in 2016 and reached 7.2 Exabytes (7.2×10^8 bytes) per month at the end of 2016, up from 4.4 Exabytes per month at the end of 2015 [2]. Although game is not the only source of mobile data traffic, how to offload data traffic of game content downloading is still an important issue since it may negatively affect players' gaming experience. Thus, it is crucial to reduce the mobile data traffic required in game contents downloading and updating. Although there are many affordable data plans offered by mobile companies, for the game which needs to frequently download or update detailed graphic maps, there is still a considerable cost on data. From the report [3], like Spacetime Studios Pocket Legends, for example, a mobile player usually uses 17MB Internet data while continuously playing for 10 minutes. Thus, the mobile player may think that it does not worth to update or download game content via cellular network. Therefore, enabling a player to download or update maps in a ubiquitous manner without monetary cost is desired. Last but not the least, 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 commutation resource provisioning for mobile players downloading game contents via wireless cellular networks also needs to be addressed when launching mobile cloud games.

Recently, the Device-to-Device (D2D) communication is proposed as a promising cooperative communication network paradigm in next generation cellular technologies, and it has drawn significant attentions in the research community. In D2D communication, nearby mobile devices in a close proximity can directly communicate to each other to share digital files and relay the data from a cellular base station to the cell-edge users. The D2D network features the high spectrum utilization, reliable communications, powering saving, overall throughput and bandwidth efficiency improvement [4]. Due to these advantages, it has been shown that D2D communication can effectively offload the mobile cellular networks to D2D networks [5] and considered as one of promising technologies for the 5G wireless communication system. The D2D communication can be used in many fields such as traffic offloading, public safety, local-based applications, and social services [6]. In this paper, we apply a hybrid network consisting both cellular and 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 reduce the waiting time when the player moves to the next scene and thus improve players' gaming experience. In mobile scenario, mobile players having unlimited data plan may enable map preloading functionality for their games.

However, it is not the case when mobile players need to pay for it. 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' downloading data rate, the limited storage size on player's device, and the 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 consider a map sharing and preloading scheme that mobile game players are in a hybrid of cellular networks and D2D networks to increase both players' utility and their offloaded data traffic. Specifically, gamers are able to preload game maps by prediction before they are entering the next game scene map.
- Different from existing works and our previous paper [7], we take the network provisioning issue into account by associating mobile players to nearby base stations and reserving bandwidth for them in case that the set of preloaded maps does not contain the entering one.
- We propose an algorithm with low computational complexity combined with a base station selection scheme for the formulated map-selection problem.
- Extensive simulations show the performance of the proposed algorithm. Simulation results will show that the utility of all mobile players can be significantly increased by the proposed algorithm.

The rest of this paper is organized as follows. In Section II, we present the related works in literature. In Section III, we setup the network and present our utility model. In Section IV, we propose the algorithm to select maps that should be downloaded *in priori* from cloud server and get best data rate for each user to download the map if it is not successfully preloaded. In Section V, we provide the simulation results for the algorithm. The paper is concluded in Section VI.

Notations: In this paper, the following notations are adopted: $\boldsymbol{X}^T$ represents the transpose of matrix $\boldsymbol{X}$; $\mathbb{R}$ is the set of real numbers, $\mathbb{R}^{m\times n}$ represents the set of $m\times n$ real matrices. $\mathbb{B}^{m\times n}$ denotes the set of $m\times n$ binary matrices. $\mathbf{1}_n$ denotes the $n\times 1$ all-one vector.

II. RELATED WORK

A. D2D Content Sharing in Cellular Traffic Offloading

Content sharing based on human users' preference via D2D network has emerged as an effective way to address traffic offloading problem, and this idea has been deeply explored by many research works. In [8], the author investigated an incentivization mechanism design issue for the D2D content-sharing communication where base station provides an offer with potential benefit to ask content owners to help forward the content for maximum base station's utility. To better estimate the preference, the historical content records are analyzed and the content preference is modeled as the probability that user may request [9][10][11]. In [9], the author proposed an algorithm to solve the problem of maximum cellular traffic offloading in an interference-aware communication model. A novel cooperative content download-and-share scheme was proposed in [12] to stimulate the demand of D2D communication usage by a simple pricing model in cellular networks in order to offload traffic burden from base stations. The work in [13] investigated a mobile cloud to self-organize D2D networks and offers efficient distribution of popular content.

TABLE I
THE TOLERANCE OF LATENCY IN ONLINE GAMES [15]

Model	Perspective	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

B. D2D in Delay Tolerant Networks

A delay-tolerant network is a network designed to operate effectively over extreme distance, such as those encountered in space communications or on an interplanetary scale [14]. This latency sometimes will be measured in hours or days. Especially, in the game scenario, although the amount of acceptable latency varies by game type, the overall criteria of latency are more strict than others. In Table I, it summarizes the maximum latency that a player can tolerate in his quality of experience (QoE) is degraded. For example, a first-person shooter (FPS), like Call of Duty, requires less than 100 millisecond latency, otherwise it may negatively affect player's gaming experience. Under these criteria, it may not work to download game contents like maps in D2D networks. However, if a map that a player has high probability to go for next movement is currently available on the player's neighborhood, the player can preload the map without monetary cost and use the cached map in the next scene without suffering any latency caused by map downloading.

In our previous paper [7], we proposed a map preloading scheme via D2D communication to maximize each game player's utility. Different from our previous work, this paper will consider the optimization problem for all game players. We propose the base station pre-selection and wireless resource provisioning for gaming content downloading to further improve palyers' gaming experience if the entering map is not preloaded.

III. SYSTEM MODEL AND PROBLEM OVERVIEW

In this section, we present our system model and give a high level overview of the problem that we need to solve in this paper. In general, we prefer to offload cellular data traffic by allowing game users to preload game scene maps from other users nearby via D2D communication but not via the wireless cellular links from the game server. In some cases, a device needs to cache selected maps locally. Taking advantage of its limited storage capacity, our objective is to maximize both game user's utility and the data traffic offloaded from game server.

A. System Overview

The proposed system is designed particularly for multiple scene games. A player in a certain proximity can access other players' cached scene maps and also can download other scene maps from neighboring players via D2D communication links. With D2D content sharing, game players can act as map senders and receivers. Specically, two players in a certain proximity can access each other player's cached scene maps. This will relieve the burden of server, save the energy consumption, and reduce the 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 users neighborhood, it is a non-trivial problem to select a set of maps and decide to 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 all players into different types based on their personalities in game. Then, we can apply the big data analysis to determine the aforementioned probabilities in a statistical manner. The main goal is to maximize the total utility for all 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: the user's downlink data rate, the caching space available on user's device, the availability of maps on user's neighbors, the user's valuation of gaming experience, the map size, and the user's affordability for cellular data traffic. Fig. 1 illustrates an example of the proposed system of map preloading. There are six mobile game players, Player $1-6$ (u_1 to u_6), and 5 base stations are available to connect, base station $1-5$ (b_1 to b_5) in this example. Two circles represent the valid D2D communication ranges of Player u_1 and Player u_4, respectively. This will allow them to access some maps from other game players within the circle for free. For example, since neighbor Player u_2 and u_3 are within the valid D2D network range of u_1, he can thus access all available maps from Players u_2 and u_3 in the neighborhood. However, Player u_1 cannot obtain the maps cached by Player $4-6$. If a certain map which is not available in neighborhood but needs to be preloaded, Player u_1 and Player u_4 can download the map from cloud server via the base station b_1 and b_2, respectively. In case that the scene that the play actually going to is not preloaded, the user needs to directly download it from the wireless cellular network. In that case, the base station selection will be an important issue for the player's gaming experience. Thus, to avoid the unpredictable latency in traditional wireless cellular networks, the provisioning of the base station association and bandwidth allocation for mobile players is necessary so that a good gaming experience can be maintained for mobile players.

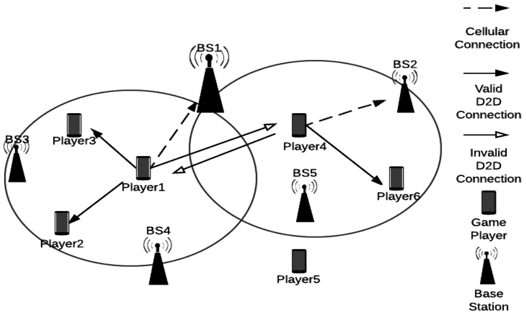

Fig. 1. An Example of Preloading Map

B. System Model

For the notational simplicity, 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 as $\mathcal{U} = \{1, \ldots, U\}$. The set of base stations is denoted as $\mathcal{B} = \{1, \ldots, B\}$. 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 $\mathcal{U}$ may have already cached several maps but 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 $u \in \mathcal{U}$ as $\mathcal{M}_u$. Thus, for a game with a large number of maps, we typically have $\mathcal{M}_u \subset \mathcal{M}$ and the following inequality holds:

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

Furthermore, in this paper, we consider the scenario that if two game players are in close proximity for a D2D communication, they can communicate directly in a D2D manner. One player can retrieve the content from other player via D2D communication. 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 monetary cost occurs when a gamer $u \in \mathcal{U}$ downloads a map $m \in \mathcal{M}$ directly from game server.

Now, we introduce the model of transitions between different scenes in the game. A player at certain time period can only be in one scene map and only one map is in use. Sooner or later, the game player will go to the next map. However, we do not know which map that the player will go *in priori* beforehand due to limited information. To tackle this problem, we need to estimate the value of each map, which is same as predicting the probability of a map that the player would go to after leaving the current map. As we consider the behaviours of game players only in a short time and we know the map where the gamer currently is, historical data analysis can provide us a good predication about the game players' future movements.

We now introduce column vector $\boldsymbol{p}_{u,m} \in \mathbb{R}^{M \times 1}$ as the game scene transmission probabilities for player $u \in \mathcal{U}$ who is currently in map $m \in \mathcal{M}$ to another map. In particular, the nth ($n \in \mathcal{M} \setminus \{m\}$) element in vector $\boldsymbol{p}_{u,m}$, denoted by $p_{u,m,n}$, represents the probability that the game player u will go to map $n \in \mathcal{M}$ after leaving the current map m. Thus, a valid vector $\boldsymbol{p}_{u,m} \in \mathbb{R}^{M\times 1}$ 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 $\boldsymbol{x}^T$ denotes 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. Channel Model

Data rate is an essential factor in our considered mobile gaming system. It will be mainly used to model the following two aspects: 1) The time saving benefit if a map is preloaded and ready to be used when the gamer entering the next scene. 2) The potential loss on preloading the game scene maps which are eventually not used. This variable can be obtained from base station, and it depends on how much bandwidth is allocated to that user. Fortunately, we find out Channon's Channel Capacity can determine the transmission data rate from certain base station. Let $r_{u,b}$ denote the data rate for game player $u \in \mathcal{U}$ connected with base station $b \in \mathcal{B}$. It is obtained as:

$$r_{u,b} = w_{u,b} \times \log_2(1 + \frac{S_{u,b}}{N_{u,b}}), \tag{2}$$

where $w_{u,b}$ is the bandwidth that base station $b \in \mathcal{B}$ allocates to user $u \in \mathcal{U}$. $\frac{S_{u,b}}{N_{u,b}}$ represents the received signal-to-noise ratio (SNR) at user $u \in \mathcal{U}$ when he is associated with base station $b \in \mathcal{B}$. Then the SNR can be expressed as:

$$\frac{S_{u,b}}{N_{u,b}} = P_b \times \frac{|h_{u,b}|^2}{\sigma^2}, \tag{3}$$

where P_b denotes the transmit power of base station b and σ^2 is the noise power at the requester, which is the game player in our scenario. $|h_{u,b}|^2$ represents the instantaneous channel gain while the average channel gain can be expressed as $E(|h_{u,b}|^2) = 1/d_{u,b}^{3.8}$ to capture the path loss effect, where $d_{u,b}$ is the distance between base station b and the game player u. We define $v_{u,b}$ as follows:

$$v_{u,b} \triangleq \frac{1}{\log_2(1 + \frac{S_{u,b}}{N_{u,b}})}, \tag{4}$$

basing on $(4) - (6)$, the bandwidth assigned to game player u from base station b can be calculated as:

$$w_{u,b} = r_{u,b} v_{u,b} o_{u,b}. \tag{5}$$

Where $w_{u,b}$ denotes the bandwidth shared to user u from base station b. Note that $o_{u,b}$ is a binary number as $o_{u,b} \in \{0, 1\}$, which is defined as follows:

$$o_{u,b} = \begin{cases} 0, & \text{if } u \text{ is not associated with base station } b, \\ 1, & \text{if } u \text{ is associated with base station } b. \end{cases} \tag{6}$$

We now introduce column vector $\boldsymbol{o}_u \in \mathbb{B}^{B\times 1}$ as the connection between player $u \in \mathcal{U}$ and base stations. It is worth mentioning that every user can only connect one base station. Thus, the valid vector $\boldsymbol{o}_u \in \mathbb{B}^{B\times 1}$ for player u satisfies $\mathbf{1} \cdot \boldsymbol{o}_u^T = 1$, which is $\sum_{b\in\mathcal{B}} o_{u,b} = 1$. Therefore, the data rate of the user r_u can be represented as follows:

$$r_u = \sum_{b\in\mathcal{B}} r_{u,b}. \tag{7}$$

Furthermore, the bandwidth allocation constraint is formulated as:

$$\sum_{u\in\mathcal{U}} w_{u,b} \le w_b, \tag{8}$$

where the sum of all the shared bandwidth $w_{u,b}$ to users $u \in \mathcal{U}$ from the same base station b should be less than or equal to the base station's total bandwidth w_b.

D. Expected Benefit Model

When a game scene transition happens, if the mobile user has already cached the new 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, 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 [16]. This will decrease the player's gaming experience.

To better serve 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:

$$i_{u,m} \triangleq S_m / r_u, \tag{9}$$

In conclusion, the time saving benefit can be written in the following form:

$$i_{u,m} = \begin{cases} \frac{S_m}{r_u}, & \text{if } m \in \mathcal{L}_u, \\ 0, & \text{if } m \notin \mathcal{L}_u, \end{cases} \tag{10}$$

where $\mathcal{L}_u$ is the map set that we have preloaded.

E. Monetary Cost Model

The game player can get the map to preload freely if it is available in the player's neighborhood. However, if the map is not available in the neighborhood, the player may still want to preload the map from cloud server via cellular networks. In this situation, a monetary cost will occur. Therefore, the monetary cost can be determined as long as the game player wants to preload some maps that are unavailable nearby. We define it as:

$$c_{u,m} \triangleq S_m q_u, \tag{11}$$

where q_u is the unit price (cent/MB) of cellular 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:

$$c_{u,m} = \begin{cases} 0, & \text{if } m \in \bigcup_{v \in \mathcal{N}_u} \mathcal{M}_v, \\ S_m q_u, & \text{if } m \notin \bigcup_{v \in \mathcal{N}_u} \mathcal{M}_v. \end{cases} \tag{12}$$

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 $J_{u,m}$ for user u brought by map m can be easily determined as $\alpha_u b_{u,m} - c_{u,m}$, where α_u is the money that a gamer would like to pay for a certain time period of playing game, and this varies from person to person. For example, for the people who chase perfect gaming experience and do not care about money, this value will relatively much higher than normal people's. However, this value will be relatively low for the people who do not enjoy playing games. While, if user $u \in \mathcal{U}$ has preloaded map $m \in \mathcal{M}$ but he does not go to it, the utility $J_{u,m}$ is $-c_{u,m}$. Such a utility will be used to weight the value of each map for map preloading selection.

F. Potential Time Loss Model

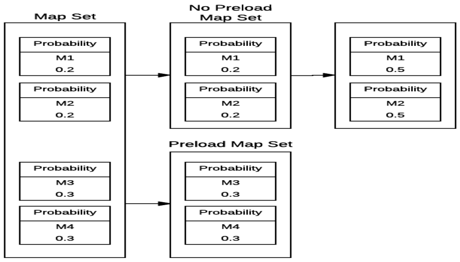

Fig. 2. An Example of Calculating Expected Loss Probability

Although we preload many game maps to maximize the utility for each gamer, there is the possibility that the gamer goes to the map which is not in gamer's preloading map set. For the sake of reality, we will calculate the *expected* map loss. Let us take the situation shown in Fig. 2, we consider there are 4 maps that are available to be preloaded for a certain gamer, but we assume that the gamer only caches M_3 and M_4 because of limited storage size. Therefore, M_1 and M_2 will be used to calculate *expected* loss. The posterior probability that the player enters M_1 and M_2 given that M_3 and M_4 are wrongly preloaded will be enlarged proportionally. Like M_1 and M_2, the conditional probability of these should be original probability times $\frac{5}{2}$. The conditional probability for *expected* loss can be calculated as:

$$\begin{aligned} &\mathbb{P}_{u,g_u,z}(X = z | z \notin (\mathcal{M}_u \bigcup \mathcal{L}_u)) \\ &\qquad = \frac{p_{u,g_u,z}}{1 - \sum_{y \in (\mathcal{M}_u \bigcup \mathcal{L}_u)} p_{u,g_u,y}}. \end{aligned} \tag{13}$$

For notational simplicity, let $p'_{u,g_u,z}$ denote the value given by (13). We have also used g_u to represent the map that user $u \in \mathcal{U}$ is currently staying at. Next, we consider if user u does not go to one of the preloaded map $m \in \mathcal{M}$, the loss received by user u in terms of the loading time can be defined as follows:

$$l_{u,m} = \frac{S_m}{r_u}. \tag{14}$$

In particular, $l_{u,m}$ represents the time used by player $u \in \mathcal{U}$ to download the entering map $m \in \mathcal{M}$ if m is not preloaded. This can be easily converted to loss utility for user u as $\alpha_u l_{u,m}$.

IV. PROBLEM FORMULATION AND PROPOSED SOLUTION

A. Problem Formulation

Our purpose is to maximize all players' *expected* utility by designing the best preloading strategy. In the end of Section III-D and Section III-F, 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 movement. Without loss of generality, we consider a set of maps $\mathcal{L}_u \subset \mathcal{M}$ that user u_x would like to preload besides those maps that user u has already cached $\mathcal{M}_u$, and let $\mathcal{T}_u$ represent the set of maps which will not be preloaded except those that user u has already cached. Denote g_u as the map that user u is currently staying at, and m as the map that user will preload and n as the map not being preloaded. The expected utility of the player is given by

$$\sum_{m \in \mathcal{L}_u} (p_{u,g_u,m} \alpha_u b_{u,m} - c_{u,m}) - \sum_{n \in \mathcal{T}_u} p'_{u,g_u,n} \alpha_u l_{u,n} \tag{15}$$

Note that for each user u, (15) depends on both the set of preloading maps $\mathcal{L}_u$ and the set of not preloading $\mathcal{T}_u$, and they satisfy $\mathcal{T}_u = \mathcal{M} \backslash (\mathcal{M}_u \bigcup \mathcal{L}_u)$. For notational simplicity, we introduce the following function for user $u \in \mathcal{U}$:$f_u(o_u, w_u, \mathcal{L}_u)$. Now, the value of (15) can be simply written as f_u. Our problem for user u can thus be formulated as follows:

$$\underset{\boldsymbol{O}, \boldsymbol{W}, \mathfrak{L}}{\text{maximize}} \sum_{u \in \mathcal{U}} f_u(\boldsymbol{o}_u, \boldsymbol{w}_u, \mathcal{L}_u), \tag{16a}$$

$$\text{subject to} \sum_{m \in \mathcal{L}_u} S_m \leq C_u - \sum_{m \in \mathcal{M}_u} S_m, \forall\, u \in \mathcal{U}, \tag{16b}$$

$$\sum_{u \in \mathcal{U}} w_{u,b} \leq w_b, \forall\, b \in \mathcal{B}, \tag{16c}$$

$$\sum_{b \in \mathcal{B}} o_{u,b} = 1, \forall\, u \in \mathcal{U}, \tag{16d}$$

$$o_{u,b} = \{0, 1\}, \forall\, u \in \mathcal{U},\, b \in \mathcal{B}, \tag{16e}$$

where the matrix $\boldsymbol{O} \triangleq [\boldsymbol{o}_1\, \boldsymbol{o}_2\, \ldots\, \boldsymbol{o}_U]$, the column vector $\boldsymbol{W} \triangleq [\boldsymbol{w}_1\, \boldsymbol{w}_2\, \ldots\, \boldsymbol{w}_B]$, and $\mathfrak{L} \triangleq \{\mathcal{L}_1, \mathcal{L}_2, \ldots, \mathcal{L}_U\}$. The constraints of Problem (16) means: 1) 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. 2) The sum of bandwidth $w_{u,b}$ shared to a group of users who connect to the base station b should less than or equal to the total bandwidth w_b of base station b. 3) A user can only connect one base station. 4) $o_{u,b}$ is a binary number.

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

```
Input: data rate $r_u$
Output: preloading map set $\mathfrak{L}$ and utility $J$
 1: Pre-calculate the ratio $J_{u,m}/S_m$.
 2: Sort the maps by the ratio $J_{u,m}/S_m$ in vector $\boldsymbol{p}_{u,m}$ in a descending
    order.
 3: Initialize $\mathcal{L}_u$ as empty for every user $u$.
 4: Set $i = 1$ as the map index.
 5: loop
 6:     for $i \in \mathcal{M}$ do
 7:         if $J_{u,t(i)} < 0$ then
 8:             Jump over the loop and stop preloading.
 9:         end if
10:         if $\varrho_u + S_{t(i)} < C_u$ then
11:             Jump over the loop but continue preloading.
12:         end if
13:         Receive the positive $J_{u,t(i)}$ and preload map $i$.
14:     end for
15: end loop
```

B. Proposed Solution

It is easy to show that Problem (16) is an NP-complete problem. Due to the limited space, we omit the proof. In this section, we come up to a solution based on the greedy algorithm. We jointly consider the availability of the maps in neighborhood, game scene transition probabilities, map size, players' downlink data rate, players' valuation on gaming experience, and "willingness to pay" of the user (*i.e.,* the affordability of the user for cellular data traffic) in the algorithm design. Since base station selection will influence the total utility for all gamers and it is difficult to directly come out a best solution, we design to use iteration to determine the best total utility. Firstly, we will solve the data rate maximization problem to get $\boldsymbol{O}$ distribution and $\boldsymbol{W}$, the problem can be formulated as follows:

$$\underset{\boldsymbol{O},\boldsymbol{W}}{\text{maximize}} \sum_{u\in\mathcal{U}} r_u, \tag{17a}$$

$$\text{subject to} \sum_{u\in\mathcal{U}} w_{u,b} \leq w_b, \forall\, b \in \mathcal{B}, \tag{17b}$$

$$\sum_{b\in\mathcal{B}} o_{u,b} = 1, \forall\, u \in \mathcal{U}, \tag{17c}$$

$$o_{u,b} = \{0,1\}, \forall\, u \in \mathcal{U},\, b \in \mathcal{B}. \tag{17d}$$

After solving the above problem, the $\boldsymbol{W}$ and $\boldsymbol{O}$ distribution will be used to get the initial data rate based on Equation (5). Then, we will call Algorithms (1) and (2) so on so forth until 90% of maps from preloading map set are the same as the previous to update best data rate and receive new preloading map set, total utility and $o_{u,b}$. The iterative calling of Algorithms (1) and (2) has been specified in Algorithm (3).

As we see from Algorithm (1), data rate can update the preloading map set $\mathfrak{L}$ and total utility J from all game player. When we update a preloading map set $\mathfrak{L}$, it may come out a new data rate r_u by calling Algorithm (2). After several iterations, if the preloading map set is convergence to a certain map set, we will assume it is the best solution for our problem.

Algorithm 2 : Based Station Association and Bandwidth Allocation for Provisioning Purpose

```
Input: preloading map set $\mathfrak{L}$
Output: data rate $r_u$
 1: while $\boldsymbol{O}$ and $\boldsymbol{W}$ are not same as previous $\boldsymbol{O}$ and $\boldsymbol{W}$ do
 2:     Compute $\boldsymbol{O}$ by mixed-integer linear programming to maximize the
        total utility $J$.
 3:     Compute $\boldsymbol{W}$ by linear programming to maximize the total utility $J$.
 4: end while
 5: Set $i = 1$ as the user $u$ index.
 6: for $i \in \mathcal{U}$ do
 7:     Set $k = 1$ as the base station $b$ index .
 8:     for $k \in \mathcal{B}$ do
 9:         if $o_{i,k} = 1$ then
10:             Compute $r_i$ with $w_{i,k}$ by equation (5).
11:         end if
12:     end for
13: end for
```

Algorithm 3 : Overall Algorithm to Solve Problem (16)

```
1: Solve problem (17) to get $\boldsymbol{O}$ distribution and $\boldsymbol{W}$.
2: Call equation (5) to get data rate $r_u$.
3: Initialize $\mathfrak{L}_p$ as empty to store previous preloading map set.
4: while the previous preloading map set $\mathfrak{L}_p$ is not 90% same as current
   preloading map set $\mathfrak{L}$ do
5:     $\mathfrak{L}_p = \mathfrak{L}$.
6:     Use data rate $r_u$ in Algorithm 1 to obtain the preloading map sets $\mathfrak{L}$
       and total utility $J$.
7:     Use $\mathfrak{L}$ as input to call Algorithm 2 to determine the data rate $r_u$.
8: end while
9: Get the best utility $J$.
```

Our map preloading selection and utility calculation algorithm is presented in Algorithm (1). We denote $J_{u,m}$ as the utility brought by map $m \in \mathcal{M}$ to user $u \in \mathcal{U}$, $t(i)$ as the index of map in the ranked column vector sorted by the ratio of $J_{u,m}/S_m$ and ϱ_u as the current left size of cache of user u, respectively. Algorithm (1) is used to determine the preloading map set and total utility, and it requires data rate r_u as input. In particular, we first pre-calculate the ratio of until to map size and sort the map set by a descending order (Lines 1-2). This can guarantee that we will use limited cache size to store the most valuable maps. Note that $i \in \mathcal{M}$ denotes the index of map in the sorted vector of $\boldsymbol{p}_{u,m}$, and define one-to-one mapping $t \triangleq \mathcal{M} \mapsto \mathcal{M}$, where $t(i)$ denotes the index of the corresponding map in set $\mathcal{M}$. Note that, there are two conditions to stop preloading: 1) All maps left are negative although we may still have spare storage (Lines 6-7). 2) We finish all storage and the left is not enough for any map with a positive utility (Lines 9-10).

Algorithm (2) illustrates the method that we use preloading map set $\mathfrak{L}$ as input to get maximize data rate r_u. We should notice that the inequality of Problem (8) is a multivariate linear programming problem that we cannot solve these two variables simultaneously. Furthermore, the $\boldsymbol{O}$ and $\boldsymbol{W}$ that bring the most utility cannot be easily determined by one time. Thus, we firstly design the iteration criteria to determine the best $\boldsymbol{O}$ and $\boldsymbol{W}$ (Line 1). Then compute $\boldsymbol{O}$ distribution by mixed-integer linear programming (Line 2) and $\boldsymbol{W}$ by linear programming (Line 3) by maximizing the total utility J based on current preloading map set. Then basing on $\boldsymbol{O}$ distribution and $\boldsymbol{W}$, we can update the data rate by Equation (5) (Lines 5-14).

Algorithm (3) is specified for interative calling of Algorithm (1) and (2), and it determines the criteria of iteration. Firstly, we solve data rate maxmization Problem (17) to get $\boldsymbol{O}$ distribution and $\boldsymbol{W}$, and then use them to get a data rate for each game player from Equation (5) (Lines 1-2). Next, circulate to call Algoriithm (1) and Algorithm (2) until the preloading map set contains 90% same maps as previous preloading map set (Lines 4-9). After iteration, we will get the best utility.

V. SIMULATION RESULTS AND DISCUSSIONS

A. Experimental Setup

In this section, the performance in our proposed scheme is shown using computer simulations. In this simulation, we consider a single cell with several MBSs (Macro Base Station) randomly distributed inside the area. The transmit power of MBS is 46 dBm. The other parameters adopted for simulation are summarized in Table II. The algorithm that we proposed in this paper will be used to compare to the algorithm that we proposed in [7], which is that game player will preload maps by the descending order of probability. The method that player only preloads available maps from neighbor gamers by a descending order or probability will also used to compare and justify the performance of the algorithm we proposed. We will introduce three kinds of map preloading techniques for performance. We refer Proposed Scheme as the algorithm we proposed and mainly focus in this paper. Random Probability Preload Scheme refer to the algorithm that we preload maps by the descending probability order and obtain data rate from random base station selection policy. For Random Server Preload Scheme refer to the algorithm that we preload maps by the descending probability order but we access all maps from server, and the data rate is obtained from random base station selection policy.

TABLE II
SIMULATION PARAMETERS

Number of Game Players	20
Valid Radius for Connection	15 m
Area	2500 m^2
Parameter in Rayleigh Distribution of Map Size	σ = 5 MB
Cache Size Average	150 MB
Data Charge	0.7 cent/MB
Ratio of Money to Time	0.33
Bandwidth	20 MHz
Default Data Rate	2.4375 MB/s
Pass Loss Exponent	3.8

B. Data Rate Performance

As we can see from Fig. 3, we firstly observate the initial data rate obtained from our proposed algorithm is very close to the data rate got from closest base station selection policy [17]. The data rate got from proposed algorithm increases when we start first several iterations and it gradually convergences to 3.75 MB/s, which is improved about 33.9% to the data rate got from closest base station selection policy.

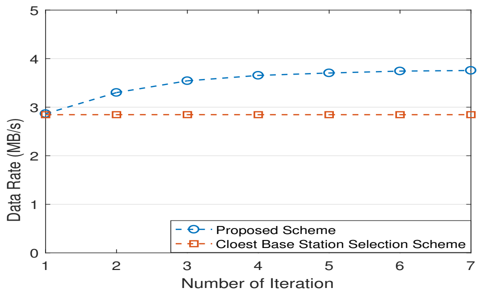

Fig. 3. Data Rate as a Function of Number of Iteration

C. Effect on Data Charge

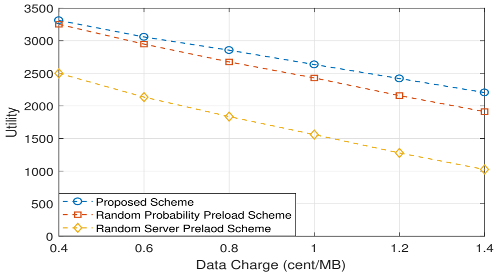

Fig. 4. Avg. Total Utility as a Function of Data Unit Price

Since the unit price of cellular data service is linearly related to monetary cost $c_{u,m}$, we expect that the total utility will drop linearly with the increase in data charge q_u. The algorithm we used in [7] brings a similar utility than the one we proposed in this paper, but it has a larger negative effect on utility with the change in price. In the algorithm that we proposed in this paper, we put every most valuable map in our map sets in utility. This can help me distinguish that some maps have high probability but bring quite less utility because of its large map size and unavailability nearby. Furthermore, it always provides a best bandwidth distribution to maximize data rate for each game player, and it drops the utility in some extend. Therefore, obviously, our new algorithm brings more utility to all game players and it has less effect on total utility.

D. Effect on Cache Size

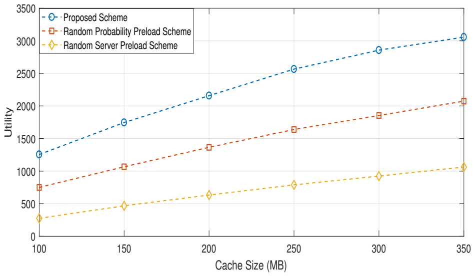

Fig. 5. Avg. Total Utility as a Function of Cache Size

Fig. 5 shows the impact of cache size on total utility performance. The first main observation is that the total utility increases almost linearly but increasing rate gradually decreases as the size of cache increases. This observation is expected as we first preload those maps brought the most utility. Especially, the proposed algorithm preloads the map with highest ratio of utility to size *in prior*. And then it preloads those maps brought positive utility to each game player until we left maps with negative expected utility or run out of cache. Comparing to the proposed algorithm, the probability based map selection algorithm has less accuracy to preload the most valuable maps under limited cache storage.

E. Effect on Bandwidth

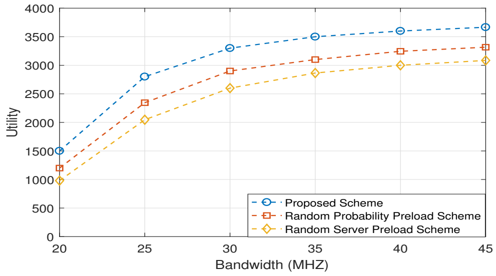

Fig. 6. Avg. Total Utility as a Function of Bandwidth

Fig. 6 illustrates the effects of bandwidth on the total utility performance. We can firstly observe that graph shows an upper bound given by the simulation result. That is, with the increase in bandwidth of each base station, the competition of bandwidth between users decreases significantly, and thus they can obtain a faster data rate. Since Data rate is used to calculate the waiting time of downloading maps that the gamers are going to, we would expect to see a decrease in waiting time loss when players get faster downloading speed. However, since data rate is in the denominator of expected loss calculation $l_{u,m}$, the decreasing rate of waiting time loss will gradually decrease and converge to some value. Therefore, we expect to see all 3 algorithms converge. The proposed algorithm has a better performance of utility improvement and the gaps between other algorithms keep slightly increase.

VI. CONCLUSION

In this paper, we have proposed an idea about preloading game maps via D2D communication to maximize the total utility for all game players. In the process of solving map selection problem, we extended the idea involving base station selection and bandwidth allocation issues. We first introduced the idea of map sharing and preloading via D2D communication network, and then proposed to use probability based on game players' behaviours to better predict the next movement. Furthermore, we formulated the map preloading and base station selection problems and combined them as an optimization problem by maximizing the total utility of all game players. In this process, a map preloading selection and data rate maximization algorithm was proposed, which can solve the problem as two subproblems. We solved the first subproblem as a mix-integer linear programming problem by using data rate. The second subproblem of getting data rate could be solved by each $o_{u,b}$ as input. Simulation results showed that our map preloading selection and data rate maximization algorithm can effectively improve each user's data rate when they downlaod from cloud server and total utility for all users, and each user can benefit from it. In addition, simulation results demonstrated the effectiveness of iteration for data rate and it showed a convergence performance of our proposed bandwidth allocation algorithm. For future work, we will consider the interrupt between users under D2D and cellular communication networks.

VII. ACKNOWLEDGEMENT

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

REFERENCES

[1] Nbn Inc, "How much data does gaming use? A handful of popular examples," Dec. 2016. [Online] Available:http://www.nbnco.com.au/blog/entertainment/how-much-data-does-gaming-use-through-popular-examples.html.

[2] Cisco 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] Z. Ashpari, "Which smartphone apps are the biggest data hogs?," Mar. 2012. [Online] Available: http://www.pcworld.com/article/252009/which-smartphone-apps-are-the-biggest-data-hogs-.html.

[4] The University of Texas at Austin, "Device-to-Device (D2D) communication: Fundamentals with applications to LTE," Jun. 2014. [Online] Available: https://wncg.org/research/briefs/ device-device-d2d-communication-fundamentals-applications-lte.

[5] 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/.

[6] M. Hicham, N. Abghour, and M. Ouzzif, "Device-to-Device (D2D) communication under LTE-advanced networks."

[7] Z. Lin, Z. Wang, W. Cai, and V. Leung, "A smart map sharing and preloading scheme for mobile cloud gaming in D2D networks," in *2017 IEEE International Conference on Smart Computing (SMARTCOMP)*, pp. 1–6, May. 2017.

[8] S. Huang, C. Yi, and J. Cai, "A sequential posted price mechanism for D2D content sharing communications," in *2016 IEEE Global Communications Conference (GLOBECOM)*, pp. 1–6, Dec. 2016.

[9] J. Jiang, S. Zhang, B. Li, and B. Li, "Maximized cellular traffic offloading via Device-to-Device content sharing," *IEEE Journal on Selected Areas in Communications*, vol. 34, pp. 82–91, Jan. 2016.

[10] Z. Wang, H. Shah-Mansouri, and V. W. S. Wong, "How to download more data from neighbors? A metric for D2D data offloading opportunity," *IEEE Transactions on Mobile Computing*, vol. 16, pp. 1658–1675, Jun. 2017.

[11] H. Jianzhang, W. Haibo, and Z. Tao, "Incentive and selection for D2D content sharing: A comprehensive scoring mechanism," in *2016 IEEE 13th International Conference on Signal Processing (ICSP)*, pp. 1208–1213, Nov. 2016.

[12] L. Fang, R. Zhang, X. Cheng, J. W. Xiao, and L. Yang, "Cooperative Content Download-and-Share (COCODaS): Motivating D2D in cellular networks," *IEEE Communications Letters*, vol. PP, no. 99, pp. 1–1, 2017.

[13] A. Moiz, M. I. Ashraf, K. S. B. Liaqat, S. Mumtaz, and M. Katz, "Implementation of D2D enabled mobile cloud based content distribution architecture in 5G networks," in *European Wireless 2016; 22th European Wireless Conference*, pp. 1–6, May. 2016.

[14] M. Rouse, "Delay-tolerant network," Mar. 2012. [Online] Available: http://searchnetworking.techtarget.com/definition/delay-tolerant-network.

[15] M. Claypool and K. Claypool, "Latency and player actions in online games," *Commun. ACM*, vol. 49, pp. 40–45, Nov. 2006.

[16] 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, pp. 5007–5021, Jul. 2016.

[17] C. Skouroumounis, C. Psomas, and I. Krikidis, "Low-complexity base station selection scheme in mmWave cellular networks," *IEEE Transactions on Communications*, to-appear.