Fifty-fourth Annual Allerton Conference
Allerton House, UIUC, Illinois, USA
September 27 - 30, 2016

Mobility-Aware Centralized D2D Caching Networks

Sameh Hosny*, Atilla Eryilmaz*, Alhussein A. Abouzeid† and Hesham El Gamal*

*The Ohio State University, USA.
Email: {hosny.1, eryilmaz.2, elgamal.2}@osu.edu

† Rensselaer Polytechnic Institute, USA. Email: abouzeid@ecse.rpi.edu

***Abstract*—The increase in demand for spectrum-based services forms a bottleneck in wireless networks. Device-to-Device (D2D) caching networks tackle this problem by exploiting users behavior predictability and the possibility of sharing data between them to alleviate network congestion. Usually, network congestion occurs at certain times of the day and in some popular locations. Consequently, the information about user's demand alone is not enough. Capturing mobility statistics allows Service Providers (SPs) to enhance their caching strategies. In this work, we introduce a mobility-aware centralized D2D caching network where a SP harnesses users demand and mobility statistics to minimize the incurred service cost through an optimal caching policy. However, the complexity of the optimal caching policy grows exponentially with the number of users. Therefore, we discuss a greedy caching algorithm which has a polynomial order complexity. We also use this greedy algorithm to establish upper and lower bounds on the proactive service gain achieved by the optimal caching policy.**

I. INTRODUCTION

The growth in data traffic represents a crucial problem in mobile networks. More than half a billion mobile devices were added in 2015 causing a 74% growth in global mobile data traffic. Nevertheless, an eightfold increase in this traffic is expected between 2015 and 2020. Moreover, three-fourths of the world's mobile data traffic will be video by 2020 [1]. This increase in demand for spectrum-based services and devices has led network SPs to experience a major demand and supply mismatch during the whole day [2]. This demand disparity is ultimately tied to user behavioral pattern. However, most people follow certain daily routines and hence their behavior is highly predictable [3],[4]. Interestingly, the time varying user activities, that are ultimately contributing to this mismatch, can be exploited to solve this demand disparity.

The concept of *proactive resource allocation* for wireless networks was established to control the supplied services to best match the demand patterns [5]. The predictability of user behavior is exploited to balance the wireless traffic over time, and significantly reduce the bandwidth required to achieve a given blocking/outage probability. *Device-to-device (D2D) communication* has been proposed in [6] as a promising technology that can relief the wireless networks congestion. A pair of end-users, moving within a close proximity to each other, establish a D2D link that can be operated in the unlicensed spectrum band, such as the Industrial, Scientific and Medical (ISM) radio bands. These D2D links when used as a traffic offloading approach introduces very little or no monetary cost for the end-users.

A tutorial overview of some recent results on base station assisted D2D wireless networks with caching for video delivery was presented in [7]. Some competing conventional schemes and a recently developed scheme based on caching at the user devices was also introduced. Throughput-outage scaling laws of such schemes were discussed. It was shown that, in realistic conditions, the D2D caching scheme largely outperforms all other competing schemes both in terms of per-user throughput and in terms of outage probability. A D2D caching network under arbitrary demand was considered in [8]. It was shown that if each node in the network can reach in a single hop all other nodes, then the proposed scheme achieves almost the same throughput of [9]. Moreover, if concurrent short range transmissions can co-exist in a spatial reuse scheme, then the throughput has the same scaling law of the reuse-only case [10], [11] or the coded-only case [9].

Two different caching scenarios have been considered in [12] where the information is either cached at a Small Base Station (SBS), or directly at the user terminals, which then use D2D communications to share the cached contents. It has been shown that, when the transmission and caching policies are jointly designed, the cache offers two possible gains, namely, the pre-downloading and local caching gains. The proposed optimal offline transmission and caching policies were used as a lower bound to evaluate the cost of any online policy. Although previous models utilized the D2D communication to alleviate network congestion, they considered a grid network formed by a set of nodes placed on a regular grid on the unit square and user's mobility was not captured in this work.

The authors in [13] considered the model of [14] and showed that the per-user throughput can increase dramatically

This work is primarily funded by the QNRF Grant NPRP 7-923-2-344, and the NSF grant CNS-WiFiUS-1456806 and 1456887. The work of A. Eryilmaz was supported by the NSF grants: CNS-NeTS-1514127, CMMI-SMOR-1562065, and CCSS-EARS-1444026; and the DTRA grant HDTRA1-15-1-0003.

when nodes are mobile rather than fixed. This improvement was obtained under several idealistic assumptions. They assumed complete mixing of nodes trajectories in the network and random mobility pattern was not considered. They also assumed that data contents are delay tolerant and stated that their ideas were not very relevant to real-time applications. Caching data contents in users devices helps us to overcome the delay constraint. Furthermore, a practical mobility model is required to represent a more realistic behavior of the users. There are many mobility models in the literature which try to capture user behavior [15]. In this work, we focus on individual user's mobility based on a probabilistic random walk and defer group mobility for our future work.

We first introduced the concept of *content trading* in D2D caching networks in [16] and more details were discussed in [17]. Users harnessed their predictable demand in proactive caching and the possibility of trading their proactive downloads to minimize their expected payments. On the other hand, SP utilized a dynamic pricing scheme to differentiate between off-peak and peak time prices and applied commissions on each trading process to maximize its profit. A novel marketplace that is based on risk sharing between users was proposed. We showed that a win-win situation is achieved between SP and end-users. However, we didn't capture in this model user's mobility which plays an important role in such marketplaces.

In this work, we consider a centralized D2D caching network where SP is aware of user's demand and mobility. We consider the results presented in this work as a forward step towards a mobile content marketplace. Our aim is to show that exploiting the information about user's mobility helps SP to optimize its caching strategy and address the network congestion problem in an intelligent manner. The main contributions are:

1) We introduce an optimal caching policy that allows SP to enhance its caching decisions based on user's demand and mobility statistics.
2) The complexity of the optimal caching policy grows exponentially with the number of users. Therefore, We introduce a sub-optimal policy based on a greedy algorithm that has a polynomial order complexity.
3) Using the proposed greedy algorithm, we establish upper and lower bounds on the gain achieved by the optimal policy for the proactive service cost.
4) The obtained results in this work allow us to enhance our content trading model presented in [16] to form a complete vision about mobile content marketplace.

The rest of the paper is organized as follows. Section II presents the proposed system model and problem formulation. SP's optimal caching policy is discussed in Section III while the proposed greedy caching policy is introduced in Section IV. Upper and lower bounds analysis for the proactive service gain achieved by the optimal policy are established in V. The paper is concluded in Section VI.

II. System Model and Problem Formulation

We consider a wireless network consisting of a set of N users $\mathcal{N} = \{1, 2, \cdots, N\}$ and a Service Provider (SP) who supplies M data items upon demand. Each data item $m \in [1, M]$ has a size $S_m > 0$ which may be a movie (as in YouTube and Netflix), a sound track (as in Panadora), a social network update (as in Facebook and Twitter), a news update (as in CNN and Fox News), etc. Each user may request any of these data items in a random fashion. We consider a time-slotted system where SP divides the duration of interest (e.g. a day) into T time slots. We assume that the duration of each slot is the time taken for a user to completely consume the requested data item and hence each time slot is in the order of minutes or possibly hours. At the beginning of each time slot, SP collects the demand of all users and supplies them with the requested data items.

A. User Demand Model

We assume that SP can track, learn and predict user behavior over time and hence constructs a *demand profile* for every user n denoted by $\mathbf{\Pi}_n = (\mathbf{p}_{n,t})_t$. For any time slot t, $\mathbf{p}_{n,t} = \left(p^m_{n,t}\right)_m$ where $p^m_{n,t}$ is the probability that user n requests item m in time slot t. The demand of user n in time slot t is captured by a random variable $\mathbb{I}^m_{n,t}$ where

$$\mathbb{I}^m_{n,t} = \begin{cases} 1, & \text{with probability } p^m_{n,t}, \\ 0, & \text{with probability } 1 - p^m_{n,t}. \end{cases}$$

We assume that at any time slot t, $\mathbb{I}^m_{n,t}$ is independent of $\mathbb{I}^m_{n,t+1}, \forall n, m$. We aslso assume that for any $n \neq k$, $\mathbb{I}^m_{n,t}$ is independent of $\mathbb{I}^m_{k,t}, \forall m, t$. Furthermore, the demand profile of each user follows a *cyclo-stationary* pattern that repeats itself in a period of T time slots. That is, we can write $\mathbf{p}_{n,t} = \mathbf{p}_{n,t+kT}$ for any non-negative integer k. As an example, the T-slot period can be interpreted as a single day through which the activity of each user varies each hour, but occurs with the same statistics every day. SP relates these time slots with the actual day time based on user's demand statistics to recognize the time slots where it experiences low demand (off-peak time) and those where a high demand occurs (peak time).

B. User Mobility Model

We assume that SP is interested in L *popular* locations $\mathcal{L} = \{1, 2, \cdots, L\}$ like airports, schools, shopping malls, stadiums or governmental buildings where high demand can be related with mobility of users. Moreover, SP can track, learn and predict the mobility of each user over time and hence constructs a *mobility profile* for every user n denoted by $\mathbf{\Theta}_n = \left(\theta^l_{n,t}\right)^l_t$ where $\theta^l_{n,t}$ is the probability that user n will be present at location l in time slot t where $\sum_{l=1}^{L} \theta^l_{n,t} = 1 \, \forall n, t$. We represent user's mobility by a modified probabilistic version of the random walk mobility model which is based on a discrete-time Markov chain model [18]. We assume that users stay in the same location within a time slot and may move to another location at the beginning of each time slot. Let $\lambda^{l,k}_{n,t}$ be the transition probability that user n moves from location l to location k in time slot t

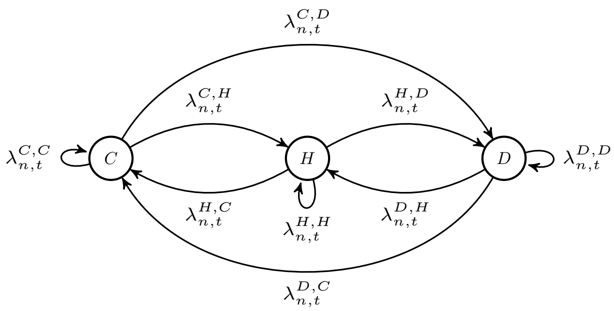


Fig. 1: An example state transition diagram of user n in time t for $L = 3$ locations.

where $\sum_{k=1}^{L} \lambda_{n,t}^{l,k} = 1 \; \forall n, t, l$. These transition probabilities may change from one time slot to another to capture the mobility of each user. However, the probability of being at a certain location in a time slot t depends on the location in the previous time slot $t-1$ only, i.e. $\theta_{n,t}^{l} = \sum_{k=1}^{L} \theta_{n,t-1}^{k} \lambda_{n,t}^{k,l}$ where $\theta_{n,1}^{l} = \lambda_{n,1}^{l,l}$.

Figure 1 shows the state transition diagram of user n in time slot t for $L = 3$ locations, like home (H), campus (C) and downtown (D). We assume that each user randomly takes a trajectory everyday starting from one location and moving to other locations. However, we assume that the mobility profile of each user follows a *cyclo-stationary* pattern that repeats itself in a period of T time slots. For example, everyday user starts from his home, visits some frequent locations throughout the day and then returns back home at the end of the day as shown in Figure 2. SP exploits this mobility to enhance its caching strategy and hence achieves more gain by reducing the incurred service cost.

C. Proactive Service Scheme

SP tries to smooth out the network load by caching some of these data items at the network edge and exploits user's mobility statistics to enhance its caching decision. We assume that one-hop device-to-device (D2D) communication is allowed and can be used to transfer data items between users. A fixed data rate link between all users is assumed. We also consider a non-fading channel between all users where an appropriate network protocol is applied to avoid multiple access interference. In the small timescale, data transmission follows an orthogonal multiple access scheme, hence inter-node interference effect is ignored in our large timescale model. For example, at location l, SP predicts that a certain item m will experience a high request in time slot t. It also predicts which users will be possibly present at that location in this time slot. This data item can be cached at these users and they can transfer it to other users in their vicinity. Therefore, some of the network load will be shifted to the D2D communication which alleviates the network congestion and yields a reduction in the incurred service cost.

Users occupy part of their device memory for caching these data items and consume some of their batteries to transfer it through the D2D communication. We capture the cost of caching each byte by a parameter $r > 0$. This parameter can be viewed as a rent cost for caching this data. We can also view it as a reward that incentives users to participate in this model and save some of their payments by getting it as a discount in their monthly bills. For simplicity, we assume that users always have enough battery level to transfer cached data items to other users in the network and that they always allow SP to cache data in their devices. This reward promotes users to raise their memory size to be able to cache more data. When N is sufficiently large, we can assume that each user has enough memory space to cache assigned data items since SP distributes cached items over all available users.

D. Cost Function

To supply requested data items, SP incurs a certain service cost due to the resources consumed at each time slot. We denote by $C(L_t)$ SP's cost for serving a total demand $L_t \geq 0$ in time slot t. We also assume that the cost function $C : \mathbb{R}^+ \to \mathbb{R}^+$ is convex and non-decreasing. We consider a *reactive network* as a baseline scenario where users' requests are served upon arrival (in contrast to proactively predicting the demand requests). In this case, the time-averaged expected cost of all users is given by:

$$C^{\mathcal{R}} = \limsup_{T \to \infty} \frac{1}{T} \sum_{t=1}^{T} \mathbb{E}\left[C\left(\sum_{n=1}^{N} \sum_{m=1}^{M} S_m p_{n,t}^{m} \right) \right]. \qquad (1)$$

where, the superscript $\mathcal{R}$ indicates reactive operation. In the proposed model, we assume that SP is aware of the demand and mobility profiles of all users over T time slots. SP caches an amount x_n^m of data item m at user n for a future possible request, i.e. user n transfers this data to other users through the D2D communication in any time slot t. SP replaces the data stored in users devices when it is expired at the end of the day (i.e. at the end of time slot T). In particular, SP caches data at the beginning of the day and lets users share throughout the rest of the day. The cached amount of data item m at each user cannot exceed its size, i.e.

$$0 \leq x_n^m \leq S_m, \; \forall n, m \qquad (2)$$

Hence, under this proactive model, the total network load in time slot t is given by (4), where the superscript $\mathcal{P}$ indicates *proactive* operation and A_n is the set of all possible combinations of n indices. i.e.

$$A_n = \Big\{ a_n := (k_1, \cdots, k_n), k_j \in \{1, 2, \cdots, N\} \; \forall j \Big\}$$

where $|A_n| = \binom{N}{n}$. We assume users share cached data items when they meet each other and get the remaining portion from SP. We also assume that users share data items between them for free. The total network load in (4) captures all cases when some of the users get together, all users get together or each user is moving alone. Consequently, the corresponding time-averaged expected cost of all users under proactive operation is given by:

$$C^{\mathcal{P}} = \limsup_{T \to \infty} \frac{1}{T} \sum_{t=1}^{T} \mathbb{E}\Big[C\Big(L_t^{\mathcal{P}} \Big) \Big] + r \sum_{n=1}^{N} \sum_{m=1}^{M} x_n^m \qquad (3)$$

$$
\begin{aligned}
L_t^{\mathcal{P}} = &\sum_{m=1}^{M}\sum_{n=2}^{N-1}\sum_{a_n\in A_n}\Big(S_m-\sum_{k\in a_n}x_k^m\Big)^{+}\sum_{k\in a_n}p_{k,t}^m\underbrace{\sum_{l=1}^{L}\prod_{k\in a_n}\theta_{k,t}^l\prod_{j\notin a_n}\Big(1-\theta_{j,t}^l\Big)}_{\text{some users are together}}\\
&+\sum_{m=1}^{M}\Big(S_m-\sum_{n=1}^{N}x_n^m\Big)^{+}\sum_{n=1}^{N}p_{n,t}^m\underbrace{\sum_{l=1}^{L}\prod_{n=1}^{N}\theta_{n,t}^l}_{\text{all users are together}}\\
&+\sum_{m=1}^{M}\sum_{n=1}^{N}\Big(S_m-x_n^m\Big)p_{n,t}^m\underbrace{\Big(1-\sum_{l=1}^{L}\sum_{k=2}^{N}\sum_{a_k\in A_k}\prod_{j\in a_k}\theta_{j,t}^l\prod_{i\notin a_k}\Big(1-\theta_{i,t}^l\Big)\Big)}_{\text{every user is alone}}
\end{aligned}
\tag{4}
$$

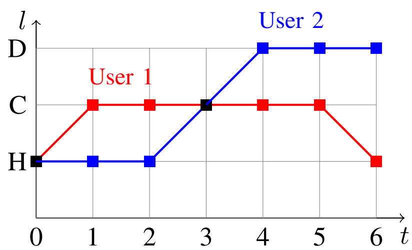


Fig. 2: An instance of the trajectory of two users for $L = 3$. They meet at home and on campus and exchange data.

E. Problem Statement

SP's gain is the difference between the *reactive* cost and the *proactive* cost under the proposed model which can be denoted by $\triangle C = C^{\mathcal{R}} - C^{\mathcal{P}}$. Users save some of their payments by finding the requested data items in their local cache or in the cache of their neighbors. SP's objective now is to achieve a positive gain (i.e. $\triangle C > 0$) by finding an optimal caching policy $\{x_n^{m*}\}_{n,m}$ which minimizes the time-averaged expected cost while delivering the requested data items on time to all users. The problem is defined as:

$$
\begin{aligned}
\min \quad & C^{\mathcal{P}}\\
\text{s.t.} \quad & (2).
\end{aligned}
\tag{5}
$$

The optimization problem in (5) depends mainly on the cost function C which may be linear, quadratic or a polynomial of higher order. The exact solution of (5) for non-linear cost functions can be obtained using convex optimization techniques. However, this case does not provide clear insights on the effect of user's mobility. Nevertheless, finding an optimal caching policy will be non-tractable. Instead, we focus here on linear cost functions to reveal some insights and to find an optimal caching policy by which allows SP to achieve a minimum service cost. The complexity of this optimal policy grows exponentially with the number of users N. We overcome this point by introducing a suboptimal policy based on a greedy algorithm which has a polynomial-order complexity. We use the sub-optimal policy to find upper and lower bounds for the optimal policy.

III. Optimal Caching Policy Analysis

In this section we introduce and optimal caching policy which achieves a minimum service cost for the proposed model. For a linear cost function, considering all possible cases of the $(.)^{+}$ terms in (4), we wind up with a set of linear programs and the optimal solution is obtained from the one which leads to a minimum cost. We start by considering two simple cases for $N = 2$ and $N = 3$ and then use them to generalize the solution.

A. Case Study: N=2

For simplicity, we start by the case when $T = 1$ and then extend it for any value of T. In this case, the suffix t can be dropped and the expected load (4) will be:

$$
\begin{aligned}
L^{\mathcal{P}} = &\sum_{m=1}^{M}\Big(S_m-\Big(x_1^m+x_2^m\Big)\Big)^{+}\Big(p_1^m+p_2^m\Big)\sum_{l=1}^{L}\theta_1^l\theta_2^l\\
&+\sum_{m=1}^{M}\Big(1-\sum_{l=1}^{L}\theta_1^l\theta_2^l\Big)\sum_{n=1}^{2}\Big(S_m-x_n^m\Big)p_n^m
\end{aligned}
\tag{6}
$$

And the optimization problem will be:

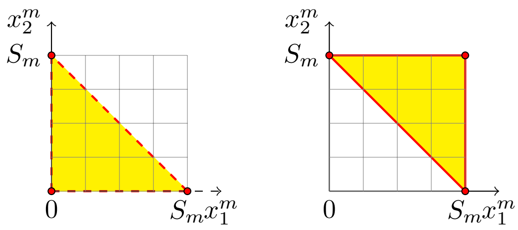
$$
\begin{aligned}
\min \quad & L^{\mathcal{P}}+r\sum_{m=1}^{M}\Big(x_1^m+x_2^m\Big)\\
\text{s.t.} \quad & 0\le x_n^m\le S_m, \qquad \forall n,m
\end{aligned}
\tag{7}
$$

The problem decomposes to M-subproblems and we have two sub-cases: either $x_1^m + x_2^m < S_m$ leading to a linear program (LP), where:

$$
\begin{aligned}
L^{\mathcal{P}} = &\sum_{m=1}^{M}\sum_{n=1}^{2}S_m p_n^m-\sum_{m=1}^{M}\sum_{n=1}^{2}x_n^m p_n^m\\
&-\sum_{m=1}^{M}\Big(x_1^m p_2^m+x_2^m p_1^m\Big)\sum_{l=1}^{L}\theta_1^l\theta_2^l
\end{aligned}
\tag{8}
$$

or $x_1^m + x_2^m \ge S_m$ which leads to another LP, where:

$$
\begin{aligned}
L^{\mathcal{P}} = &\sum_{m=1}^{M}\sum_{n=1}^{2}S_m p_n^m-\sum_{m=1}^{M}\sum_{n=1}^{2}x_n^m p_n^m\\
&-\sum_{m=1}^{M}\sum_{n=1}^{2}\Big(S_m-x_n^m\Big)p_n^m\sum_{l=1}^{L}\theta_1^l\theta_2^l
\end{aligned}
\tag{9}
$$



(a) $x_1^m + x_2^m < S_m$ (b) $x_1^m + x_2^m \geq S_m$

Fig. 3: Feasibility regions for $N = 2$

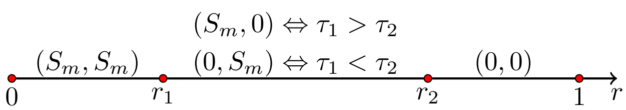


Fig. 4: Optimal Policy for $N = 2$

Note that the first term in (8) and (9) represents the reactive load of the network, the second term represents the *caching gain* achieved by caching x_1^m and x_2^m at users 1 and 2 respectively, while the last term represents the *sharing gain* attained when each user transfer his proactive download to the other. The feasibility regions of these LPs are shown in Figure 3. The optimal solution of each LP is at one of its extreme points in the corresponding feasibility region. We have 4 extreme points $(0,0), (S_m, 0), (0, S_m)$ and (S_m, S_m). The optimal solution of the problem is that of the LP which yields a minimum service cost. This solution can be extended for a general T. Figure 4 shows SP's optimal policy which is explained in the following proposition.

Proposition 1. *For $N = 2$ and for each data content m, SP's optimal caching policy follows:*

1) (S_m, S_m) *is optimal if and only if:*

$$r < r_1 = \min_i \tau_i,$$
$$0 \leq \tau_i = \frac{1}{T} \sum_{t=1}^{T} p_{i,t}^m \left(1 - \sum_{l=1}^{L} \theta_{1,t}^l \theta_{2,t}^l\right) \leq 1$$

2) *(0,0) is optimal if and only if:*

$$r > r_2 = \max_i \rho_i,$$
$$\rho_i = \frac{1}{T} \sum_{t=1}^{T} \left(p_{i,t}^m + p_{j,t}^m \sum_{l=1}^{L} \theta_{1,t}^l \theta_{2,t}^l\right) \geq 0$$

3) $(S_m, 0)$ *is optimal if and only if* $\tau_1 > \tau_2$.
4) $(0, S_m)$ *is optimal if and only if* $\tau_2 > \tau_1$.

Proof. The proof is straightforward by evaluating the cost function at all extreme points and comparing them to find the optimal solution. □

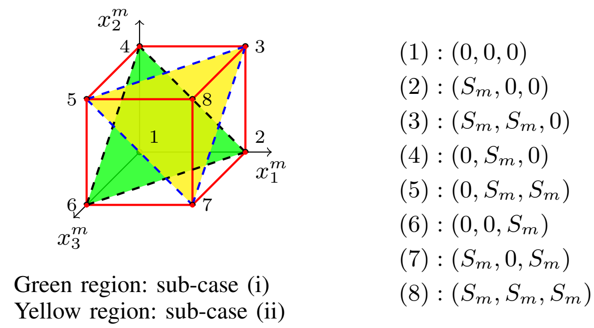


Fig. 5: Feasibility regions for $N = 3$

B. *Case Study: N=3*

For simplicity, we start by the case when $T = 1$ and then extend it for any value of T. In this case, the suffix t can be dropped and the optimization problem will be:

$$\begin{aligned} \min \quad & L^{\mathcal{P}} + r \sum_{m=1}^{M} \sum_{n=1}^{3} x_n^m \\ \text{s.t.} \quad & 0 \leq x_n^m \leq S_m, \quad \forall n, m \end{aligned} \tag{10}$$

The problem decomposes to M-subproblems and we have $1 + \sum_{n=1}^{3} \binom{3}{n} = 8$ sub-cases. Each sub-case leads to a different LP. The optimal solution is that of the one which yields a minimum service cost among all sub-cases. It is enough to consider only two sub-cases: (i) when all terms inside the $(.)^+$ functions are positive leading to a LP, where:

$$\begin{aligned} L^{\mathcal{P}} = & \sum_{m=1}^{M} \sum_{n=1}^{3} S_m p_n^m - \sum_{m=1}^{M} \sum_{n=1}^{3} x_n^m p_n^m \\ & - \sum_{m=1}^{M} \sum_{i \neq j} \left(\left(x_i^m p_j^m + x_j^m p_i^m \right) \sum_{l=1}^{L} \theta_i^l \theta_j^l \right) \end{aligned} \tag{11}$$

(ii) when all the terms inside the $(.)^+$ functions are negative and can be removed leading to another LP, where:

$$\begin{aligned} L^{\mathcal{P}} = & \sum_{m=1}^{M} \sum_{n=1}^{3} S_m p_n^m - \sum_{m=1}^{M} \sum_{n=1}^{3} x_n^m p_n^m \\ & - \sum_{m=1}^{M} \sum_{n=1}^{3} \left(S_m - x_n^m \right) p_n^m v_n \end{aligned} \tag{12}$$

where,

$$v_i = \sum_{l=1}^{L} \left(\theta_i^l \theta_j^l + \theta_i^l \theta_k^l \left(1 - \theta_j^l\right) \right), i \neq j \neq k$$

Note that the first term in (11), (12) represents the reactive load of the network, the second term represents the *caching gain* achieved by caching x_1^m, x_2^m and x_3^m, while the last term represents the *sharing gain* attained when each user transfer his cached data to other users. The feasibility regions of these LPs are shown in Figure 5. Each LP has 4 extreme points and its solution is one of them. The optimal solution of the problem is the solution of the LP which yields a minimum service cost. This solution can be extended for a general T.

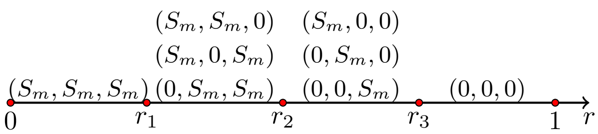


Fig. 6: Optimal Policy for $N = 3$

Figure 6 shows SP's optimal policy which is explained in the following proposition.

Proposition 2. *For $N = 3$ and for each data content m, SP's optimal caching policy follows:*

1) (S_m, S_m, S_m) *is optimal if and only if:*

$$r < r_1 = \min_i \frac{1}{T} \sum_{t=1}^{T} p_{i,t}^m \Big(1 - v_{i,t}\Big), 0 \leq r_1 \leq 1$$

2) *Caching once is better than twice if and only if:*

$$r > r_2 = \max_i \left\{ \max_{k \neq j \neq i} \left\{ \frac{1}{T} \sum_{t=1}^{T} \left(p_{j,t}^m \Big(1 - \sum_{l=1}^{L} \theta_{i,t}^l \theta_{j,t}^l \Big) + p_{k,t}^m \sum_{l=1}^{L} \theta_{j,t}^l \theta_{k,t}^l \Big(1 - \theta_{i,t}^l\Big) \right) \right\} \right\}, 0 \leq r_2 \leq 1$$

3) $(0, 0, 0)$ *is optimal if and only if:*

$$r > r_3 = \max_i \frac{1}{T} \sum_{t=1}^{T} \left(p_{i,t}^m + \sum_{j \neq i} p_{j,t}^m \sum_{l=1}^{L} \theta_{i,t}^l \theta_{j,t}^l \right) \geq 0$$

4) *For the case of caching once, it is optimal to cache at user k_1 if and only if:*

$$k_1 = \arg\max_i \ \frac{1}{T} \sum_{t=1}^{T} \left(p_{i,t}^m + \sum_{j \neq i} \left(p_{j,t}^m \sum_{l=1}^{L} \theta_{i,t}^l \theta_{j,t}^l \right) \right) \tag{13}$$

5) *For the case of caching twice, it is optimal to cache at users k_1 and k_2 if and only if:*

$$(k_1, k_2) = \arg\max_{i,j} \ \frac{1}{T} \sum_{t=1}^{T} \left(p_{i,t}^m + p_{j,t}^m + \sum_{k \neq i,j} p_{k,t}^m v_{k,t} \right) \tag{14}$$

Proof. The proof is straight forward by evaluating the cost function for every extreme point and comparing them to find the optimal solution. □

C. *Optimal Policy for N-users*

Now, from previous cases we can infer the optimal caching policy for a general number of users N as shown in Figure 7. For each data content m, SP needs to find N points on the scale or r. In particular, there are $N+1$ regions on this scale starting from caching nothing up to caching everywhere. These points (regions) determine how much caching should be done based on the exact value of r. Inside each region of r, users should be ranked to determine which user caches this data content. Based on the demand and mobility profiles of all users, SP finds an optimal caching decision $\{x_n^{m*}\}_n^m$

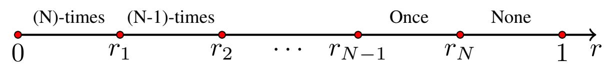


Fig. 7: Optimal Policy for N users

which minimizes the incurred service cost. We summarize SP's optimal caching policy in Algorithm 1.

Remark 1. *SP's optimal caching decision depends on the exact value of r which represents the caching cost. In particular, smaller value of r yields more caching and vice versa.*

Remark 2. *Higher values of users meeting probabilities shift points $r_1, r_2, \cdots, r_{N-1}$ to the left while point r_N moves to the right. In particular, the possibility of over-caching reduces when meeting probabilities increase. Moreover, higher demand values shift all points to the right and yields more caching.*

Proposition 3. *The complexity of SP's optimal caching policy described in Algorithm 1 grows exponentially with the number of users N.*

Proof. Although the number of points on the scale of r increases linearly with the number of users N, the total number of required terms for users ranking is given by:

$$\binom{N}{1} + \binom{N}{2} + \cdots + \binom{N}{N-2} + \binom{N}{N-1} = \sum_{k=0}^{N} \binom{N}{k} - \binom{N}{N} - \binom{N}{0} = 2^N - 2$$

which increases exponentially with N. □

IV. Greedy Cachign Policy

The complexity of optimal policy discussed in the previous section motivates us to introduce a greedy algorithm as a sub-optimal caching policy. Moreover, this algorithm allows us to establish upper and lower bounds on the achieved service gain of the optimal caching policy. The main idea of this algorithm depends on ranking users based on the level-(1) list stated in (15). Users are picked from this list in order to cache the amount required of each data content. Algorithm 2 summarizes the steps of this greedy algorithm.

Proposition 4. *Complexity of the greedy caching policy described in Algorithm 2 grows in a polynomial order with the number of users N.*

Proof. We still need to find N points the scale of r which increases linearly with the number of users. However, the total number of required terms is given by:

$$1 + 2 + \cdots + (N-2) + (N-1) + N = \frac{N(N+1)}{2}$$

□

Algorithm 1 Optimal Caching Policy for N-users

Given: $N, M, L, \mathbf{\Pi}_n, \mathbf{\Theta}_n$
for $m = 1$ to M **do** CACHING($N, L, m, p^m_{n,t}, \theta^l_{n,t}$)
procedure CACHING($N, L, m, p^m_{n,t}, \theta^l_{n,t}$)

Step 1:
Create a level-(1) ranked list using:

$$s_i = \frac{1}{T}\sum_{t=1}^{T}\left(p^m_{i,t} + \sum_{j\neq i}\left(p^m_{j,t}\sum_{l=1}^{L}\theta^l_{i,t}\theta^l_{j,t}\right)\right), \forall i \quad (15)$$

r_N is the first item in list, k_1 is its corresponding index.

Step 2:
Create a level-(2) ranked list using:

$$s_{ij} = \frac{1}{T}\sum_{t=1}^{T}\left(p^m_{i,t} + p^m_{j,t} + \sum_{k\neq i,j}\left(p^m_{k,t}\sum_{l=1}^{L}\left(\theta^l_{i,t}\theta^l_{k,t} + \theta^l_{j,t}\theta^l_{k,t} - \theta^l_{i,t}\theta^l_{j,t}\theta^l_{k,t}\right)\right)\right), \forall i, j \quad (16)$$

k_1, k_2 are the corresponding indexes of the first item. calculate r_{N-1} as follows:

$$r_{N-1} = \frac{1}{T}\sum_{t=1}^{T}\left(p^m_{k_2,t}\left(1 - \sum_{l=1}^{L}\theta^l_{k_1,t}\theta^l_{k_2,t}\right) + \sum_{n\neq k_1,k_2} p^m_{n,t}\sum_{l=1}^{L}\theta^l_{k_2,t}\theta^l_{n,t}\left(1 - \theta^l_{k_1,t}\right)\right) \quad (17)$$

⋮

Step $N-1$:
Create a level-$(N-1)$ ranked list using

$$s_i = \frac{1}{T}\sum_{t=1}^{T}\left(p^m_{i,t}\left(1 - v_{i,t}\right)\right), \forall i, \quad (18)$$

$$v_{i,t} = \sum_{l=1}^{L}\left[\sum_{i\neq j}\theta^l_{i,t}\theta^l_{j,t} - \sum_{i\neq j\neq k}\theta^l_{i,t}\theta^l_{j,t}\theta^l_{k,t}\cdots(-1)^N\prod_{n=1}^{N}\theta^l_{n,t}\right]$$

r_1 is the smallest item in list, k_N is the corresponding index and all other users cache it.

V. Upper And Lower Bounds Analysis

The greedy caching policy ranks users based on the level-(1) list stated in (15). This ranking is similar to that of caching once in the optimal policy. However, this doesn't guarantee that level-(1) ranking is still valid in all other cases. For instance, the first two users in this list are not guaranteed to be the same users for the case of caching twice in the optimal policy. Hence, the greedy caching policy forms a lower bound for the proactive service gain achieved by the optimal policy.

Theorem 1. *Under demand and mobility profiles of N-users and for $T \geq 1$, the optimal proactive service gain*

Algorithm 2 Greedy Caching Policy for N-users

Given: $N, M, L, \mathbf{\Pi}_n, \mathbf{\Theta}_n$
for $m = 1$ to M **do** CACHING($N, L, m, p^m_{n,t}, \theta^l_{n,t}$)
procedure CACHING($N, L, m, p^m_{n,t}, \theta^l_{n,t}$)

Step 1:
Create a level-(1) ranked list using:

$$s_i = \frac{1}{T}\sum_{t=1}^{T}\left(p^m_{i,t} + \sum_{j\neq i}\left(p^m_{j,t}\sum_{l=1}^{L}\theta^l_{i,t}\theta^l_{j,t}\right)\right), \forall i$$

r_N is the first item in list, k_1 is its corresponding index.

Step 2:
Exclude user k_1 and create a level-(2) ranked list using:

$$s_j = \frac{1}{T}\sum_{t=1}^{T}\left(p^m_{j,t}\left(1 - \sum_{l=1}^{L}\theta^l_{k_1,t}\theta^l_{j,t}\right) + \sum_{k\neq j} p^m_{k,t}\sum_{l=1}^{L}\theta^l_{j,t}\theta^l_{k,t}\left(1 - \theta^l_{k_1,t}\right)\right), \forall j \neq k_1 \quad (19)$$

r_{N-1} is the first item in list.
k_2 is the corresponding index.

Step 3:
Exclude users k_1, k_2 and create a level-(3) list by:

$$s_k = \frac{1}{T}\sum_{t=1}^{T}\left(p^m_{k,t}\left[1 - \sum_{l=1}^{L}\left(\theta^l_{k_1,t}\theta^l_{k,t} + \theta^l_{k_2,t}\theta^l_{k,t}(1 - \theta^l_{k_1,t})\right)\right] + \sum_{n\neq k} p^m_{n,t}\sum_{l=1}^{L}\theta^l_{k,t}\theta^l_{n,t}\left(1 - \theta^l_{k_1,t}\right)\left(1 - \theta^l_{k_2,t}\right)\right), \forall k \neq k_1, k_2 \quad (20)$$

r_{N-2} is the first item in list, k_3 is the corresponding index.

⋮

So on for all other ranges of r.

$\triangle C(\mathbf{\Pi}_n, \mathbf{\Theta}_n)$ of (5) achieved by Algorithm (1) satisfies:

$$\triangle C(\mathbf{\Pi}_n, \mathbf{\Theta}_n) \geq \triangle C_L(\mathbf{\Pi}_n, \mathbf{\Theta}_n) \quad (21)$$

where, $\triangle C_L(\mathbf{\Pi}_n, \mathbf{\Theta}_n)$ is the gain achieved by Algorithm (2).

Proof. We compare the gain achieved by the optimal and greedy policies. For example, in the case of caching once both policies achieve the same gain and we have:

$$\triangle C(\mathbf{\Pi}_n, \mathbf{\Theta}_n) = \triangle C_L(\mathbf{\Pi}_n, \mathbf{\Theta}_n) = \sum_{m=1}^{M} S_m\left[\frac{1}{T}\sum_{t=1}^{T}\left(p^m_{i,t} + \sum_{j\neq i}\left(p^m_{j,t}\sum_{l=1}^{L}\theta^l_{i,t}\theta^l_{j,t}\right)\right) - r\right] \quad (22)$$

In the case of caching twice, the optimal caching policy has a proactive service gain defined by:

$$\triangle C(\mathbf{\Pi}_n, \mathbf{\Theta}_n) = \sum_{m=1}^{M} S_m\left[\frac{1}{T}\sum_{t=1}^{T}\left(p^m_{i,t} + p^m_{j,t} + \sum_{k\neq i,j} p^m_{k,t}\sum_{l=1}^{L}\left(\theta^l_{i,t}\theta^l_{k,t} + \theta^l_{j,t}\theta^l_{k,t} - \theta^l_{i,t}\theta^l_{j,t}\theta^l_{k,t}\right)\right) - 2r\right] \quad (23)$$

this gain depends mainly on the selection of users i and j. Since level-(1) ranking can not guarantee that these users are the same users as in the greedy policy, this gain is larger than or equal to the gain achieved by the greedy caching policy. The same applies to all other cases. □

Moreover, the greedy algorithm allows us to establish an upper bound for the optimal proactive service gain. Level-(1) ranking defined in (15) generates N items representing the gain achieved by caching data content m once at one of the users. Adding these gains up provides us with an upper bound for the gain achieved by the optimal caching policy in all cases. For example, the first item is this ranked list is an upper bound for the case of caching once in the optimal policy. The sum of the first two items is an upper bound for the case of caching twice in the optimal policy and so on.

Theorem 2. *Under demand and mobility profiles of N-users and for $T \geq 1$, the optimal proactive service gain $\triangle C\left(\mathbf{\Pi}_n, \mathbf{\Theta}_n\right)$ of (5) achieved by Algorithm (1) satisfies:*

$$\triangle C\left(\mathbf{\Pi}_n, \mathbf{\Theta}_n\right) \leq \triangle C_U\left(\mathbf{\Pi}_n, \mathbf{\Theta}_n\right) \tag{24}$$

where, $\triangle C_U\left(\mathbf{\Pi}_n, \mathbf{\Theta}_n\right)$ is the gain achieved by adding up gains defined in (15).

Proof. We show this result by comparing the gain achieved by the optimal caching policy with the again achieved by the greedy caching policy by adding up items of level-(1) ranking list (15). For the case of caching once we have:

$$\triangle C\left(\mathbf{\Pi}_n, \mathbf{\Theta}_n\right) = \triangle C_U\left(\mathbf{\Pi}_n, \mathbf{\Theta}_n\right) \tag{25}$$

which is the same value as in (22). For the case of caching twice, we have: $\triangle C_U\left(\mathbf{\Pi}_n, \mathbf{\Theta}_n\right)$

$$\begin{aligned} &= \sum_{m=1}^{M} S_m \left[\frac{1}{T}\sum_{t=1}^{T}\left(\left(p_{i,t}^m + p_{j,t}^m\right)\left(1+\sum_{l=1}^{L}\theta_{i,t}^l\theta_{j,t}^l\right)\right.\right. \\ &\left.\left. + \sum_{k\neq i,j}\left(p_{k,t}^m\sum_{l=1}^{L}\left(\theta_{i,t}^l\theta_{k,t}^l + \theta_{j,t}^l\theta_{k,t}^l\right)\right)\right) - 2r\right] \end{aligned} \tag{26}$$

Comparing (26) with (23) we see that:

$$\triangle C_U\left(\mathbf{\Pi}_n, \mathbf{\Theta}_n\right) \geq \triangle C\left(\mathbf{\Pi}_n, \mathbf{\Theta}_n\right) \tag{27}$$

Same approach applies to all other cases. □

VI. Conclusion and Discussions

We considered a mobility-aware centralize D2D caching network where SP takes the caching decision based on demand and mobility statistics of users. These statistics allowed SP to predict users requests in popular locations which experience high demand volumes. Capturing mobility statistics enhanced the network performance and helped SP to minimize its incurred service cost. An optimal caching policy for a linear proactive service cost function was introduced. The complexity of the proposed policy grows exponentially with the number of users. Therefore, we introduced a sub-optimal policy based on a greedy caching algorithm with a polynomial order complexity. The greedy policy was used to establish upper and lower bounds on the proactive service gain achieved by the optimal caching policy. The results of this work extends our understanding for users behaviour in D2D caching networks and allows us to add mobility dynamics to our content trading model discussed in [16]

We considered a centralized D2D caching network, one of our future work directions is to consider distributed caching scenario to address the complexity issue in an efficient manner. It also captures users selfishness by taking a caching decision towards their payment minimization. Furthermore, we studied the impact of individual user's mobility model. However considering a group mobility model enriches the problem and adds more possibility to enhance SP's caching strategy. Group mobility models allow us to consider the social relationships between users which also play an important role in D2D caching networks.

References

[1] [Online]. Available: Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2015–2020 White Paper

[2] FCC, "Spectrum policy task force report, fcc 02-155," 2002.

[3] C. Song, Z. Qu, N. Blumm, and A.-L. Barabási, "Limits of predictability in human mobility," vol. 327, no. 5968, pp. 1018–1021, 2010.

[4] K. Farrahi and D. Gatica-Perez, "Discovering human routines from cell phone data with topic models," in *Wearable Computers, 2008. ISWC 2008. 12th IEEE International Symposium on*. IEEE, 2008, pp. 29–32.

[5] J. Tadrous, A. Eryilmaz, and H. El Gamal, "Proactive resource allocation: Harnessing the diversity and multicast gains," *Information Theory, IEEE Transactions on*, vol. 59, no. 8, pp. 4833–4854, Aug 2013.

[6] C.-H. Yu, K. Doppler, C. B. Ribeiro, and O. Tirkkonen, "Resource sharing optimization for device-to-device communication underlaying cellular networks," *Wireless Communications, IEEE Transactions on*, vol. 10, no. 8, pp. 2752–2763, 2011.

[7] G. C. M. Ji and A. F. Molisch, "Wireless device-to-device caching networks: Basic principles and system performance," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 1, pp. 176–189, 2016.

[8] M. Ji, G. Caire, and A. F. Molisch, "Fundamental limits of caching in wireless d2d networks," *IEEE Transactions on Information Theory*, vol. 62, no. 2, pp. 849–869, Feb 2016.

[9] M. A. Maddah-Ali and U. Niesen, "Fundamental limits of caching," *IEEE Transactions on Information Theory*, vol. 60, no. 5, pp. 2856–2867, May 2014.

[10] M. Ji, G. Caire, and A. F. Molisch, "Optimal throughput-outage tradeoff in wireless one-hop caching networks," in *Information Theory Proceedings (ISIT), 2013 IEEE International Symposium on*, July 2013, pp. 1461–1465.

[11] ——, "The throughput-outage tradeoff of wireless one-hop caching networks," *IEEE Transactions on Information Theory*, vol. 61, no. 12, pp. 6833–6859, Dec 2015.

[12] M. Gregori, J. Gómez-Vilardebó, J. Matamoros, and D. Gündüz, "Wireless content caching for small cell and d2d networks," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 5, pp. 1222–1234, May 2016.

[13] M. Grossglauser and D. N. C. Tse, "Mobility increases the capacity of ad hoc wireless networks," *IEEE/ACM Transactions on Networking*, vol. 10, no. 4, pp. 477–486, Aug 2002.

[14] P. Gupta and P. R. Kumar, "The capacity of wireless networks," *IEEE Transactions on Information Theory*, vol. 46, no. 2, pp. 388–404, 2000.

[15] T. Camp, J. Boleng, and V. Davies, "A survey of mobility models for ad hoc network research," *Wireless Communications and Mobile Computing (WCMC)*, vol. 2, no. 5, pp. 483–502, Aug 2002.

[16] F. Alotaibi, S. Hosny, J. Tadrous, H. E. Gamal, and A. Eryilmaz, "Towards a marketplace for mobile content: Dynamic pricing and proactive caching," *arXiv preprint arXiv:1511.07573*, 2015.

[17] F. Alotaibi, S. Hosny, H. E. Gamal, and A. Eryilmaz, "A game theoretic approach to content trading in proactive wireless networks," in *2015 IEEE International Symposium on Information Theory (ISIT)*, June 2015, pp. 2216–2220.

[18] C.-C. Chiang, "Wireless network multicasting," Ph.D. dissertation, University of California, Los Angeles, 1998.