Interference-Aware Distributed Device-to-Device Caching

Ahmet Cihat Kazez*, Tolga Girici*,
*TOBB University of Economics and Technology
Department of Electrical and Electronics Engineering
06561, Ankara, Turkey
Email: {ackazez,tgirici}@etu.edu.tr

***Abstract*—In this paper, cache placement problem in cellular networks with device to device communications (D2D) is addressed. D2D transmission allows cellular bandwidth reuse which necessitates interference management. We propose a distributed algorithm, where the base station rewards the user terminals for caching content and helping other nearby terminals. Each user terminal distributively decides which contents to cache in a way that maximizes its reward and minimizes the interference to the base station. The base station decides the optimal reward in a way that minimizes its cost. The proposed method is compared with baseline caching schemes. Numerical results reveal that the proposed method both achieves less base station cost and interference than the benchmarks.**

I. Introduction

With recent developments in the wireless communication systems and smartphone technology, there has been an increasing interest for enriched content such as video streaming. This phenomenon results in a dramatic growth in the cellular traffic load. Deploying more base stations, increasing the amount of bandwidth and improvements in the physical layer wireless communication technology can be proposed as a solution to this problem. However, at certain locations and at certain times of the day peak cellular load may reach to unsupportable amounts. Caching the enriched content such as video streaming at the network edge, such as base stations, helpers and devices is an attractive solution in order to alleviate the peak load [1]. With this solution, previously demanded content can be cached at the edge nodes (i.e. base station and devices) and these nodes can directly supply the future demands without increasing the backhaul traffic. Caching at the devices can be of extra help, since the cached content can be transmitted to the nearby devices thanks to device-to-device (D2D) transmission. This type of transmission can be performed with low power, due to the short proximity. Moreover, in D2D transmission devices can reuse the cellular bandwidth which improves spectral efficiency significantly. In this scenario, interference due to frequency reuse has to be managed intelligently.

D2D cache management is a challenging problem that has been studied intensively in the recent literature. For instance, [2] analyzes the interference scaling by using stochastic geometry and finds the optimal D2D collaboration distance in order to control the interference. The authors in [3] study the throughput scaling of D2D caching in the presence of coded caching. In coded caching the devices transmit the linear combinations of packets in their cache library. In [4] authors look at the throughput-outage tradeoff and show that a D2D caching network is scalable, as the per user throughput is constant as the network grows. Throughput scaling can be made even better by allowing multihop transmission [5]. Scaling results and performance limits of D2D caching networks are summarized in [6].

The group of works mentioned above mainly considered the limiting behavior of networks for some basic caching and transmission schemes. However optimizing the caching policy is also important. Two important metrics are hit probability (the chance of finding a nearby device having the requested content) and throughput (which also depends on the *success* of the actual transmission). The authors in [7], [8] address a stochastic network and derive the hit probability and average throughput and then use them in optimizing the probabilistic caching policy. In [9] effects of mobility on the hit and success probability is investigated and it is observed that a heavy tailed file size distribution may be better for the success probability. In [10] optimal caching decision is found in a way that minimizes latency. The authors in [11] propose a local caching algorithm and a subsequent D2D matching algorithm. The authors in [12] consider a hotspot area and search for an optimal clustering of D2D users, where users in the same cluster can share cached contents. In [13] caching is done at the small base stations, but the caching decision is made based on social D2D interactions of users. The work in [14] tries to find the optimal amount of helpers in a D2D network. In [15] the authors consider a dynamic system model, where the devices request contents and they may choose to cache them in order to help other users in the future. They don't take into account interference and the propose an online policy that is far from the optimal offline policy. Another dynamic system is considered in [16], where a single cluster of D2D users is studied. The authors optimize distributed storage coding in order to maximize energy efficiency.

There are a group of works such as [17], [18] that study optimal link scheduling and/or power allocation in cellular networks that perform D2D caching. The work in [17] also takes into account contact duration of nearby users. These works do not propose a caching policy.

In a cellular system of large number of nodes, it may be better to give caching decisions in a distributed manner. Stackelberg game is a useful framework for distributed optimization algorithms. In this scheme there is a leader (base station) and followers (user terminals). The leader announces a price or a reward and the followers optimize their decisions in order to maximize their net utility. Then the leader can update the price/reward in order to reach an optimal operating point. This framework is being used in the D2D caching literature. For example in [19] the BS gives a reward to users for helping others and the users maximize their net utility (received reward minus delay cost). The BS minimizes the reward it gives plus the service cost, which is the amount of content that it cannot offload to devices. In [20] the mobile network operator (MNO) is the leader and the content providers (CPs) are the followers. The MNO sells storage space to CPs and the CPs optimize the stored contents in order to maximize the satisfaction of its users [21].

In this work we propose a distributed algorithm for cache placement in cellular systems with D2D transmissions. We assume D2D transmissions that reuse the cellular bandwidth, which necessitates interference management. Our algorithm encourages caching of popular content by the user terminals that would cause less interference to the base station.

II. System Model

In our model network, a set $\mathcal{N} = \{1, 2, \ldots, N\}$ of user terminals (UTs) randomly distributed in a sector of a circular cellular area. Let $\mathcal{N}_n \subseteq \mathcal{N}$ be the set of neighbors of UT n. Each UT can form D2D link with its neighbors. For simplicity we assume that two nodes n, m are neighbors if their distance $d_{n,m}$ is less than d_0. Let $h_{n,m}$ be the channel gain from node n to m, which incorporates pathloss, shadowing and fading. We also assume that D2D transmissions reuse the uplink bandwidth.

Each UT n has a cache capacity of C_n and can cache any content from the set of available contents $\mathcal{C} = \{1, \ldots, C\}$. Without loss of generality we assume that all contents are of equal size and coded using rateless coding (e.g. LT or Raptor Codes). According to this assumption a UT can cache a fraction of a content. Let $0 \leq x_n^c \leq 1$ denote the fraction of content c cached by UT m. A UT requesting content c can retrieve fractions of it from multiple UTs and if the total fraction reaches 1, the content can be decoded. If the total amount of cached content is less than 1, then the UT can retrieve the rest of it from the BS.

Cache management mechanism depends on the content popularity. Let p_n^c be the popularity of content c at UT n. This is also the probability of content c is requested by UT n hence $\sum_{c\in\mathcal{C}} p_n^c = 1$. Content popularities are modeled by a Zipf distribution and may vary for different UTs. When UT n requests content c it first looks at its own cache. If it has the full content, then no transmission is required. Else, if $x_n^c < 1$, then UT n asks its neighbors in the order of increasing distance, until the decodability threshold is satisfied.

Let $\gamma_{n,m}$ be the signal to noise ratio (SNR) of transmission from UT n to m,

$$\gamma_{n,m} = \frac{P_{n,m} h_{n,m}}{\sigma^2} \tag{1}$$

Here $P_{n,m}$ is the transmission power and σ^2 is the noise power. We assume that for a successful transmission SNR should be greater than γ_0. Therefore the required transmit power for successful transmission from UT n to m becomes,

$$P_{n,m} = \frac{\gamma_0 \sigma^2}{h_{n,m}} \tag{2}$$

As we mentioned before, each UT orders its neighbors in decreasing order of channel gain. Let $(k)_n$ be the k^{th} best neighbor of UT n. Obviously the best neighbor of a UT is itself $((0)_n = n)$. Let $[n]_m$ is the rank of UT n among the neighbors of UT m. The fraction of content c that UT n transmits to a requesting UT m is as follows,

$$F_{n,m}^c(x_n^c, \mathbf{x}^c) = \min\left\{x_n^c, \max\left\{0, 1 - \sum_{k=0}^{[n]_m - 1} x_{(k)_m}^c\right\}\right\} \tag{3}$$

Since UTs reuse the cellular bandwidth, they cause interference at the BS while transmitting cached content to other nearby UTs. Let $h_{n,0}$ be the channel gain between UT n and the BS. We formulate the average interference created by UT n as follows,

$$I_n(\mathbf{x}) = \sum_{c\in\mathcal{C}} \sum_{m\in\mathcal{N}} p_m^c h_{n,0} \frac{\gamma_0 \sigma^2}{h_{n,m}} F_{n,m}^c(x_n^c, \mathbf{x}^c) \tag{4}$$

This expression may not be the actual instantaneous interference but it gives a good reflection of it. UTs which are too close to the BS and have distant D2D links create more interference.

The aim of D2D caching is offloading traffic from the BS. If the total portion of a content obtained from the neighbors is less than 1, the residual content has to be supplied by the BS. The average amount of total content that BS has to supply is formulated as follows,

$$F_0(\mathbf{x}) = \sum_{c\in\mathcal{C}} \sum_{n\in\mathcal{N}} p_n^c \max\{0, 1 - \sum_{m\in\mathcal{N}_n} x_m^c\} \tag{5}$$

This is called the service cost. Decreasing the service cost requires increasing the D2D caching. On the other hand, increased D2D transmissions would cause interference to the BS. Since service cost and interference are conflicting aims a careful caching decision has to be made by UTs. According to We will propose a distributed scheme, where each UT determines its own caching.

III. Stackelberg Game Formulation

BS pays a reward (incentive) r to UTs proportionally to the amount of unit content provided to its neighbors. For UT n the net utility is the incentive received from the BS, minus the average interference it creates at the BS while serving to its neighbors.

$$\max_{\mathbf{x}_n} U_n(r, \mathbf{x}_n, \mathbf{x}) = \sum_{c\in\mathcal{C}} \sum_{m\in\mathcal{N}_n} p_m^c r s_c F_{n,m}^c(\mathbf{x}^c) - w_d I_n(\mathbf{x}) \tag{6}$$

s.t

$$\sum_{c\in\mathcal{C}} x_n^c s_c \leq C_n \tag{7}$$

$$\sum_{n\in\mathcal{N}} I_n(\mathbf{x}) \leq I^{max} \tag{8}$$

Here I^{max} is the maximum total interference constraint. When this constraint is violated, BS signals the UTs and no more content is cached.

The BS aims to minimize its cost, which is the total reward (incentive) paid to UTs and the service cost, due to content requests which cannot be offloaded.

$$C(r, \mathbf{x}) = \sum_{c\in C} \sum_{n\in\mathcal{N}} p_n^c s_c \sum_{m\in\mathcal{N}_n} r F_{m,n}^c(\mathbf{x}^c) + w_s F_0(\mathbf{x}) \tag{9}$$

Algorithm 1 states the proposed reward-based distributed cache placement algorithm.

Algorithm 1 Stackelberg Game-Based Caching

1: Initialize reward $r = 0$.
2: **while** $C(r, \mathbf{x})$ is decreasing **do**
3: $r = r + \Delta_r$
4: Initialize cache states $x_n^c = 0, \forall n \in \mathcal{N}, c \in \mathcal{C}$.
5: **while** Converge= 0 and $\sum_{n\in\mathcal{N}} I_n(\mathbf{x}) \leq I^{max}$ **do**
6: **for** $n \in \mathcal{N}, c \in \mathcal{C}$ **do**
7: $\nabla_n^c(r, \mathbf{x}) = \frac{U(r, x_n^c+\delta, \mathbf{x}) - U(r, x_n^c, \mathbf{x})}{\delta}$
8: **end for**
9: **if** $\sum_{c\in\mathcal{C}} x_n^c + \nabla_n^c(r, \mathbf{x}) \times \gamma \leq C_n$ **then**
10: Update $x_n^c = x_n^c + \nabla_n^c(r, \mathbf{x}) \times \gamma \forall n \in \mathcal{N}, c \in \mathcal{C}$
11: **end if**
12: **end while**
13: Compute $C(r, \mathbf{x})$
14: **end while**

BS initializes reward to zero at the beginning (Line 1). The main loop (Lines 2-14) increases the reward in Δ_r steps. Increasing reward triggers an increase in the cached content which reduces the serving cost of BS. In other words, more and more traffic offloaded to D2D transmission. At a certain point, increasing the reward more does not decrease service cost as much as it increases the reward cost, where the total BS cost $C(r, \mathbf{x})$ starts to increase. At this point the algorithm stops.

In the inner while loop (Lines 5-12) each device independently calculates a gradient, $\nabla_n^c(r, \mathbf{x})$ for each content. This basically the amount of utility increase by increasing the caching of a content by an amount of δ. Amount of reward is announced by the BS at each stage. Increasing caching improves the reward of a UT, while it also increases the interference created at the BS. Therefore this scheme encourages caching of the popular contents and caching by the UTs that won't cause much interference at the BS. At one point caching more content does not increase the reward as much as it increases the interference cost, or the device fills up its cache capacity, where the device stops incrementing its cached contents. At this point the device may send a signal to the BS. The inner loop also terminates when the total average interference to the BS reaches the limit I^{max}.

In order to implement this algorithm in a distributed manner, some information exchange between nodes is necessary. In this manner, before algorithm starts, BS has to serve content popularities to UTs. Also, during the algorithm, each UT m has to receive feedback from its neighbors n about the interference that they create at the BS ($h_{n,0}\frac{\gamma_0\sigma^2}{h_{n,m}}$) in order to handle interference constraint and each node has to inform its neighbors about its caching status.

IV. Numerical Results

A. Simulation Setup

In simulations, a network model with a BS, $N = 8$ D2D enabled UTs and $C = 20$ contents is considered. UTs are distributed in a 60^o sector of a cell. According to our assumption, all contents and UTs' caches are of the equal size, where $s_c = 1, \forall c \in \mathcal{C}$ and $C_n = 2, \forall c \in \mathcal{C}$ respectively.

D2D pathloss is assumed to be $148 + 40\log_{10}(d_{n,m})$, where $d_{n,m}$ is the D2D link distance in kilometers. UT to BS pathloss is assumed to be $128.1 + 36.7\log_{10}(d_n)$, where d_n is the distance between UT n and BS.

TABLE I: List of Simulation Parameters

Notation and Value	Description
$N = 8$	Number of UT's
$C = 20$	Number of Contents
$rad = 0.2km$	Radius of the area
$s_c = 1$	Size of c^{th} content
$C_n = 2$	Cache size of n^{th} UT
$r = [0 - 1]$	Reward interval
$\Delta_r = 0.02$	Step size of reward increase
$\delta = 0.001$	Step size of the cached content increase in the Stackelberg approach
$\gamma = 0.001$	Gradient multiplier in the Stackelberg approach
$\alpha = 1$	Skewness of the Zipf distribution
$w_r = 3$	Weight of the reward cost
$w_i = 2$	Weight of the interference
$\beta = 0.005$	Step size of the cached content increase in the Baseline algorithms

B. Baseline Algorithms

In order to evaluate the performance of Stackelberg Game-Based caching we also analyse the following baseline caching schemes [19] in terms of cost and interference. In these baseline schemes UTs incrementally update their initially empty cache states until the cache size or interference constraint is violated, iteratively depending on the following procedures.

In each iteration, each UT;

1) Random Caching (RC): Randomly selects a content and add β amount of it to its cache.

2) Uniform Caching (UC): Adds β amount of every content to its cache.
3) Popularity Based Caching (PBC): Adds βp_n^c amount of every content to its cache. Where p_n^c is the preference of content c by n^{th} UT.
4) Greedy Caching (GC): Adds β amount of its favourite content to its cache. When the favourite content is fully cached, the UT starts the next favourite content.

Simulations are performed for 150 random network topologies. For each topology the distributed algorithm and the baseline methods are run. Resulting total cost and the average total interference to the BS for each topology is recorded.

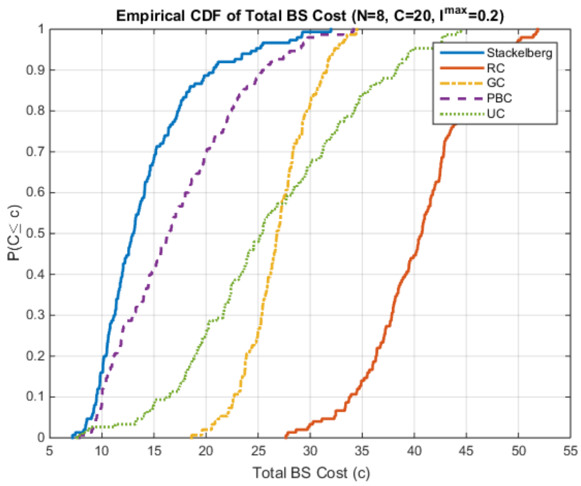

Fig. 1: Cumulative distribution function for Base station's total cost, for maximum interference limit of $I_{max} = 0.2$.

Figure 1 shows the empirical cumulative distribution function of the service cost for the five methods and for $I^{max} = 0.2$. In terms of the median service cost the algorithms are ordered as Stackelberg, PBC, UC, GC and RC. In terms of 90-percentile service cost they are ordered similarly as Stackelberg, PBC, GC, UC and RC. Popularity-based caching is the best benchmark, but our algorithm achieves 20% better service cost.

Figure 2 shows the empirical cumulative distribution function of the average total interference for the five methods where $I^{max} = 0.2$. As seen in the result, all the baseline schemes fill up the interference quota I^{max} more than half of the cases. On the other hand, median interference for the proposed algorithm is five times less than I^{max}.

Figure 3 shows the empirical cumulative distribution function of the total cost for the five methods where $I^{max} = 0.25$. With this relaxed interference constraint, the cost performance of Popularity Based Caching approaches to that of proposed algorithm. When we look at the interference distribution in Figure 4, we see that the proposed algorithm still causes significantly less interference at the BS with respect to baseline caching schemes.

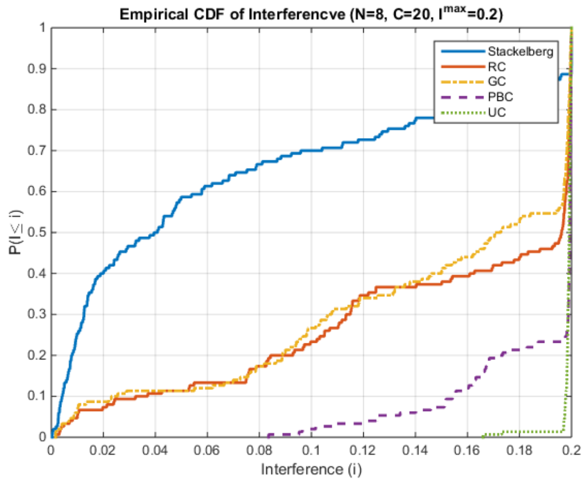

Fig. 2: Cumulative distribution function for average total interference to the Base station, for maximum interference limit of $I_{max} = 0.2$.

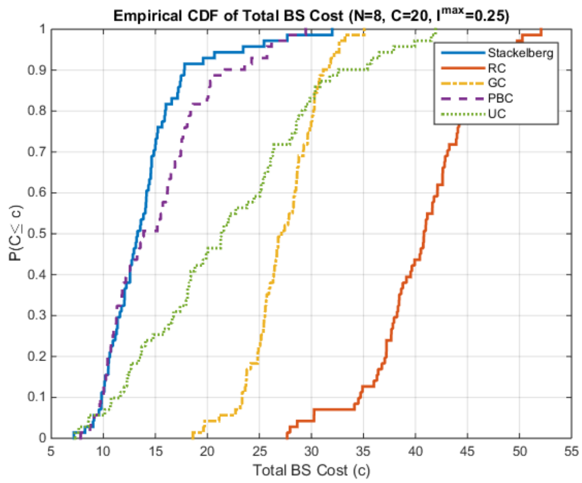

Fig. 3: Cumulative distribution function for Base station's total cost, for maximum interference limit of $I_{max} = 0.25$.

V. Conclusion

In this work we addressed the problem of optimal and distributed cache placement in the user terminals in the presence of D2D transmission. Unlike the literature we take into account the interference caused at the BS by the D2D transmissions. The proposed reward based distributed cache placement scheme successfully offloads traffic to the user terminals. The proposed algorithm both achieves better offloading and results in significantly less interference to the BS, than the baseline algorithms.

For the future work we plan to formulate cache placement as a centralized optimization problem that minimizes service cost subject to a total average interference constraint. This will serve as a benchmark for our proposed distributed algorithms.

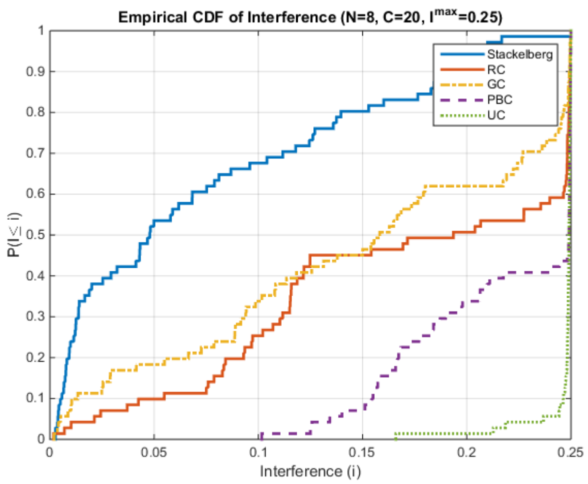

Fig. 4: Cumulative distribution function for average total interference to the Base station, for maximum interference limit of $I_{max} = 0.25$.

VI. Acknowledgements

This research is supported by the Turkish Scientific and Technological Research Council (TÜBİTAK) Project No. 215E123.

References

[1] E. Bastug, M. Bennis, and M. Debbah, "Living on the edge: The role of proactive caching in 5g wireless networks," *IEEE Communications Magazine*, vol. 52, no. 8, pp. 82–89, Aug 2014.

[2] N. Golrezaei, A. G. Dimakis, and A. F. Molisch, "Scaling behavior for device-to-device communications with distributed caching," *IEEE Transactions on Information Theory*, vol. 60, no. 7, pp. 4286–4298, July 2014.

[3] M. Ji, G. Caire, and A. F. Molisch, "Fundamental limits of distributed caching in d2d wireless networks," in *2013 IEEE Information Theory Workshop (ITW)*, Sept 2013, pp. 1–5.

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

[5] S. W. Jeon, S. N. Hong, M. Ji, G. Caire, and A. Molisch, "Wireless multihop device-to-device caching networks," *IEEE Transactions on Information Theory*, vol. PP, no. 99, pp. 1–1, 2017.

[6] M. Ji, G. Caire, 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, Jan 2016.

[7] Z. Chen, N. Pappas, and M. Kountouris, "Probabilistic caching in wireless d2d networks: Hit optimal vs. throughput optimal," *IEEE Communications Letters*, vol. PP, no. 99, pp. 1–1, 2016.

[8] J. Rao, H. Feng, C. Yang, Z. Chen, and B. Xia, "Optimal caching placement for d2d assisted wireless caching networks," in *2016 IEEE International Conference on Communications (ICC)*, May 2016, pp. 1–6.

[9] C. Jarray and A. Giovanidis, "The effects of mobility on the hit performance of cached d2d networks," in *2016 14th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt)*, May 2016, pp. 1–8.

[10] H. Hsu and K. C. Chen, "A resource allocation perspective on caching to achieve low latency," *IEEE Communications Letters*, vol. 20, no. 1, pp. 145–148, Jan 2016.

[11] 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, no. 1, pp. 82–91, Jan 2016.

[12] B. Chen, C. Yang, and G. Wang, "Cooperative device-to-device communications with caching," in *2016 IEEE 83rd Vehicular Technology Conference (VTC Spring)*, May 2016, pp. 1–5.

[13] E. Baştuğ, M. Bennis, and M. Debbah, "A transfer learning approach for cache-enabled wireless networks," in *2015 13th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt)*, May 2015, pp. 161–166.

[14] A. Zhang, L. Wang, and L. Zhou, "Location-based distributed caching for device-to-device communications underlaying cellular networks," *Wireless Communications and Mobile Computing*, 2015.

[15] 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.

[16] J. Pääkkönen, A. Barreal, C. Hollanti, and O. Tirkkonen, "Coded caching clusters with device-to-device communications," *arXiv preprint arXiv:1605.09002*, 2016.

[17] H. Wu, L. Wang, T. Svensson, and Z. Han, "Resource allocation for wireless caching in socially-enabled d2d communications," in *2016 IEEE International Conference on Communications (ICC)*, May 2016, pp. 1–6.

[18] L. Zhang, M. Xiao, G. Wu, and S. Li, "Efficient scheduling and power allocation for d2d-assisted wireless caching networks," *IEEE Transactions on Communications*, vol. 64, no. 6, pp. 2438–2452, June 2016.

[19] Z. Chen, Y. Liu, B. Zhou, and M. Tao, "Caching incentive design in wireless d2d networks: A stackelberg game approach," in *2016 IEEE International Conference on Communications (ICC)*, May 2016, pp. 1–6.

[20] F. Shen, K. Hamidouche, E. Baştuğ, and M. Debbah, "A stackelberg game for incentive proactive caching mechanisms in wireless networks," *arXiv preprint arXiv:1609.02596*, 2016.

[21] Q. Wang, W. Wang, S. Jin, H. Zhu, and N. T. Zhang, "Quality-optimized joint source selection and power control for wireless multimedia d2d communication using stackelberg game," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 8, pp. 3755–3769, 2015.