Manipulation with Domino Effect for Cache- and Buffer-Enabled Social IIoT: Preserving Stability in Tripartite Graphs

Document Version:
Peer reviewed version

Manipulation with Domino Effect for Cache- and Buffer-Enabled Social IIoT: Preserving Stability in Tripartite Graphs

Yanjing Sun, *Member, IEEE,* Bowen Wang, Song Li, *Member, IEEE*, Zhi Sun, *Member, IEEE*, Hien M. Nguyen, Trung Q. Duong, *Senior Member, IEEE*

Abstract—As a new Internet of Things (IoT) paradigm where smart devices work socially by exploiting social tie with adjacent devices, the Social IoT (SIoT) can effectively meet the real-time data sharing demands in Industrial IoT (IIoT) scenario, with the inter-device social relations being incentives. Besides, pre-caching on device level can potentially combat the backhaul capacity bottlenecks. Considering the limited cache memory, we may not use the whole capacity for caching, but leave a fraction for buffering data packets. In this paper, we investigate how to maximize the quality of experience (QoE) while minimizing the energy consumption. Firstly, we design a proactive cache placement scheme for cost minimization. Next, we conceive the content sharing procedure with the framework of tripartite graph and propose a ternary stable matching algorithm to let devices self-organize the content sharing. Finally, we prove that inconspicuously manipulation with domino effect can further improve the system performance.

Index Terms—Social IoT, industrial IoT, cache and buffer, tripartite graph, inconspicuously manipulation.

I. INTRODUCTION

THE Internet of Things (IoT) has shifted the paradigm of user communication towards heterogeneity and pervasiveness, where lots of data is generated and exchanged by surrounding smart objects for particular applications, such as smart manufacturing, fault diagnostics and health-care systems [1]. The deployment of IoT in industry, termed as Industrial IoT (IIoT), is an emerging domain that enables efficient and sustainable production [1]. With the increasing number of industrial sensors across physical environments, it is challenging to handle the vast amounts of real-time data generated by IIoT devices. Considering that most IIoT devices are carried by human, we can leverage their owners' social ties to improve the performance of information exchange in IIoT. Besides, by investigating the application of social networking in IoT, smart devices can establish social ties and share information socially with each other when they are in close proximity, no matter whether their owners have any social interaction or not, which conforms to the Social IoT paradigm [2]. The Social IoT is defined as A Social Collaborative Internet of Things" in [2], where smart devices can establish social ties, thereby achieving shared goals. However, the integration of Social IoT with IIoT is still in its infancy.

In IIoT scenarios such as surrounding rock structure monitoring of deep underground roadway in intelligent coal mining and fault diagnostics in industrial field, the cloud data center will collect different kinds of rough data from a mass of sensors, then perform data analysis and processing. The IIoT device requests the processed data packets from data center, and decides which operation to perform. Since self-organized decision making is the main goal in IoT, the importance of inter-device social trust cannot be ignored. The authors in [3] surveyed the recent advances and challenges in social-aware networks. This survey argued that devices may not cooperate for data sharing without any real incentives and enough social trust. The authors in [4] designed an incentive scheme for distributed computation in cyber physical systems based on social reputation. Besides the challenges in virtual social domain, there are also some bottlenecks in real physical domain. If the content delivery procedure requires downloading redundant contents repeatedly during a period, enormous pressure will be posed to the traffic load at the base stations (BSs). Intuitively, if the popular contents can be pre-stored into the intermediate nodes, known as caching, the redundant content transmission will be avoided to a certain extent [5]. In [6], the authors studied the caching in Social IoT scenario, where IoT devices can pre-cache some desired contents and share them socially with surrounding devices, which inspired us to integrate it in IIoT scenario.

As the predominant solution for seamless coverage in future heterogeneous networks, small-cell enabled architectures can bring contents closer to the CRs. However, establishing and maintaining the small base stations (SBSs) are costly [7]. For further traffic offloading, distributed caching allows pre-caching and directly delivering contents via device-to-device

Yanjing Sun, Bowen Wang, Song Li and Zhi Sun are with the School of Information and Control Engineering, China University of Mining and Technology, Xuzhou, 221116, China (e-mail: {yjsun, bowenwang, lisong}@cumt.edu.cn, zhisun@buffalo.edu).

Bowen Wang and Trung Q. Duong are with the School of Electronics, Electrical Engineering and Computer Science, Queens University Belfast, BT7 1NN, UK (trung.q.duong@qub.ac.uk).

Zhi Sun is also with the Department of Electrical Engineering, The State University of New York at Buffalo, Buffalo, NY 14260, USA.

Hien M. Nguyen is with Duy Tan University, Danang, Vietnam (e-mail: nguyenminhhien2501@gmail.com).

(D2D) communications, which means that the IoT devices can act as content helpers (CHs). The D2D caching has recently gained momentum with a high number of recent papers, covering both terrestrial D2D caching [6]–[8] and D2D caching in the sky [9], [10]. In general, the entire content sharing procedure comprises two stages: cache placement and content delivery. In the first stage, popular contents will be pre-cached during off-peak periods, and then these contents will be delivered to the content requesters (CRs) in the second stage. However, how to jointly solve both the cache placement and content delivery by exploring social attributes has not been studied in the existing studies [6]–[10]. The key challenge of D2D caching of D2D in Social IIoT is how to efficiently leverage social attributes to improve the system performance in both cache placement and content delivery phases. Since cache memory is usually limited on devices, some contents may not be pre-cached. When delivering uncached contents, the buffer can temporally store the contents as a short-term memory [11]. Considering that CHs usually have a small coverage, contents cached at these CHs (ahead of time) may not be requested by CRs within coverage (later on). In this setting, CHs will not use the whole memory space for file caching, but leave a fraction of memory for buffering data packets. The purpose of buffering is to enable adaptive link selection and to combat channel fading, which eventually enhances the transmission rate of content delivery. In [12], the authors provided a detailed performance analysis of buffer-aided relaying, and demonstrated that buffer-aided relay can achieve significant performance gains compared to conventional relaying. Considering both cache- and buffer-enabled networks, the average delay was optimized with the fixed buffer capacity constraint in [13]. In [11], the authors proposed a joint spectrum allocation and cache placement scheme to optimize the average latency considering the storage capacity constraint. However, both of these two works investigated the effect of buffer on SBS instead of on device level, and thus ignored the effect of inter-device social relations on content delivery. Considering that most mobile users (MUs) are selfish and rational, it is impractical that selected CHs are willing to consume power and storage capacity to provide cache placement and delivery services for free, which is important to achieve synergy between caching and communications.

The content sharing problem also belongs to the category of resource allocation. Intuitively, virtual resources such as videos and figures can be efficiently managed by exploiting social characteristics. Based on these characteristics, game-theocratic or graph-theocratic methods can be applied to allocate resource in centralized/distributed manners. In [14], the authors worked out the optimal solution using a three-dimensional matching algorithm based on the Hungary algorithm. However, the proposed method is centralized so that the BS may suffer from excessive load. By relaxing the stability condition, the author in [15] investigated how to find a k-ary stable matching beyond bipartite graph and proved the NP-completeness of determining the existence of stable matching, which inspired us to describe the cache- and buffer-enabled relaying networks as the tripartite graph.

However, the derived stable matching result may not meet the requirement of resource owners' (CHs in this paper) revenues. Inspired by this, some works investigated how to unilaterally improve the benefit of agents in one side by manipulating the preference profile [16]. In [17], the authors proved that manipulation in stable marriage problem is NP-hard. A coalition strategy was proposed in [18] by using the cheating strategy to benefit D2D users in heterogeneous cellular networks. However, the resulting matching is not stable with respect to the true preference after cheating so that the stability is not guaranteed. Besides, the falsified list is nearly entirely different from the true preference list of the manipulator, which makes the manipulation easy to be detected. The authors in [16] proposed an interesting standpoint that a single woman can manipulate the men-proposing Gale-Shapley (GS) algorithm in [19] to obtain the women-optimal matching result, which motived us to study the interplay of manipulators.

As discussed above, most related works on distributed caching suffer from three major issues: 1) the effect of social characteristics on distributed caching is ignored, which can provide incentive for content sharing; 2) they overlook the importance of real incentives such as social and monetary rewards to encourage more MUs to act as CHs; and 3) they focused more on the revenues of CRs instead of those of CHs. Besides, maximizing the quality of service (QoE) is generally energy costly, which encourages us to trade off both the QoE performance and energy consumption. With these issues in heart, we firstly leverage the social attributes to design the incentive mechanism, and then we integrate social attributes into matching phase, which promotes the content sharing willingness of MUs. Finally, we propose an inconspicuous manipulation strategy to unilaterally improve the benefit of CRs with the QoE level guarantee. Our contributions are fourfold:

- *Modeling*: We design a comprehensive content sharing scheme in social IIoT scenario targeting at maximizing the QoE while minimizing the energy consumption. We depict the entire content sharing scheme as a tripartite graph, thereby decoupling the optimization problem into two separate problems: proactive cache placement problem with the available cache and buffer capacity constraint, and distributed content delivery problem.
- *Algorithm*: We propose a joint cache placement and content delivery algorithm to solve above problems in two stages. In the first stage, the proactive cache placement scheme is proposed to balance cache and buffer. In the second stage, we leverage social reputation to design the incentive mechanism and propose a ternary stable matching algorithm to solve the content delivery problem in a distributed way.
- *Manipulation*: To further benefit more CHs, we study how to manipulate the CR-proposing GS algorithm to obtain the CH-optimal matching result. We first prove that the existence of domino effect in two-sided stable matching can guarantee that inconspicuously manipulating the matching result by single CH while preserving the stability is feasible.

- *Validations*: We provide comprehensive theoretical proofs to prove that our proposed algorithm has better optimality, stability, and complexity properties. Last but not least, under various simulations based on realistic data set, the proposed scheme always have superior performance compared with other state-of-art benchmark methods, which is able to compensate for the system performance degradation caused by limited storage capacity.

The rest of this paper is organized as follows. The architecture for social IIoT scenario is presented and then the optimization problem is elaborated in Section II. The scheme of joint cache placement and content delivery is proposed in Section III. The inconspicuous manipulation is described in Section IV. Simulation results are provided in Section V, followed by conclusion in Section VI.

II. SYSTEM MODEL AND PROBLEM FORMULATION

As shown in Fig.1, we consider a Social IIoT scenario consisting of a macro base station (MBS), K SBSs, and a number of IIoT devices. In this paper, we consider the application scenario such as surrounding rock structure monitoring of deep underground roadway in intelligent coal mining, the MBSs serving as cloud data center will periodically collect various rough data from sensors, perform data analysis and processing. Each SBS will pre-cache part of the data packets from MBS for traffic offloading with limited cache storage capacity [20]. The IIoT device acting as both a controller and actuator, requests the processed data packets from MBS or SBSs, and decides which operation to perform (such as sending alerts, turning on/off a pump, moving a mechanical arm to a certain position) based on the empirical dataset. For example, the IIoT device for monitoring will send alerts if the deformation of surrounding rock structure may potentially cause the collapsing based on the received empirical dataset. The total number of IIoT devices can be divided into N CHs and M CRs. Let us denote by $SBS = \{s_k\}_{k=1}^{K}$ the set of SBSs, $CH = \{h_i\}_{i=1}^{N}$ the set of CHs, $CR = \{r_j\}_{j=1}^{M}$ the set of CRs. Both CHs and CRs need to request processed data packets for service execution. In such a three-tier caching scenario, the MBS as cloud data center will first push contents into SBSs to reduce the traffic load. Since caching at SBSs sometimes suffers from longer latency and cannot cover all the cell's area, the SBSs wish to employ some IoT devices, termed as CHs, for further traffic offloading. Note that the IoT devices who have high social trust and storage capacity will be selected as CR. Since cache memory is limited at user equipments, some contents may not be pre-cached and the delivery of uncached contents may lead to more delay. To overcome this performance bottleneck, CHs may not use the whole memory space for caching, but leave a fraction of memory for buffering data packets. Based on the multi-tier caching scheme in [20], we mainly consider three transmission modes: 1) D2D mode: if the requested content has been pre-cached entirely and directly transmission can provide better QoE performance, the CR will request it from CH; 2) Cellular mode: in the first case, if the D2D link fails or the CR can get better QoE by directly downloading from nearby SBS

rather than D2D link, the CR will request it from SBS. In the second case, if the CR cannot received the entire content from adjacent CHs, and the delay caused by buffer-aided relay for delivering the remaining fraction in the second stage is longer than downloading it from SBS, the CR will request it from SBS. In the third case, if adjacent SBS and CHs cannot provide the requested content, the CR will download it from MBS; 3) D2D+Relay mode: if requested content has not been pre-cached entirely and buffer-aided relay can provide better QoE performance than downloading it from SBS, the CH will firstly deliver the cached fraction of requested content, and then deliver the remaining fraction by acting as a buffer relay for this CR.

Let $F = \{1, ..., f, ..., F\}$ denote the complete file library, where each file L_f with unequal size also represents one type of tasks or applications. The popularity distribution $Q = \{q_1, ..., q_f, ..., q_F\}$ following the Zipf distribution in a certain connection period is assumed to be identical for all IoT devices, which is expressed as $q_f = f^{-\gamma} / \sum_{l=1}^{F} l^{-\gamma}$, where γ represents the Zipf parameter, referring to different popularity. This popularity represents that CRs request different contents according to their current application. The available storage capacity of CH i is denoted by C_i. Due to the limited storage capacity, each CH can only cache some of the file library. The proportion of file f cached at CH i is denoted by $x_{if} \in [0, 1]$. In this paper, we consider the scenario in which some idle users will firstly be recruited as CH and pre-cache some popular contents in off-peak period to relieve the offloading of SBSs in the peak period [5], [6]. The CRs on task will request needed data packets from CHs, SBSs or MBS, to perform corresponding operations. We also consider the self-caching for CRs and thus those CRs having cached their desired contents can act as CHs potentially.

As aforementioned, the storage capacity comprises both the cache as long term memory and buffer capacity as short-term memory. Besides, the cache chip that can store files in a long time and buffer chip that can temporally store files are interchangeable [21]. Naturally, the buffer capacity of CH i is denoted by $C_i - \sum_{f \in F} x_{if} L_f$. From Fig.1, CH i located in SBS k's coverage first catches x_{if} proportion of file f from SBS k. When file f is requested, the remaining $1 - x_{if}$ proportion will be delivered to CH i from the SBS k and then relayed to CR j from SBS k via buffering. If x_{if} equals to 1, the transmission cost of CH i will be significantly reduced via D2D link instead of relay link. Hence, the proactive caching scheme is crucial to make sure that the required contents will be always entirely pre-cached so that the cost will be reduced.

A. Physical Domain

Considering that the dynamically changing interference will affect both the cache placement and content delivery states when sharing spectrum with different users. Since bandwidth allocation problem is not the main focus in this paper, we mainly study the content sharing scheme with fixed bandwidth allocation[1]. Similarly, we adopt the method in [22], which utilizes the signal-to-interference-and-noise ratio (SINR) in

---

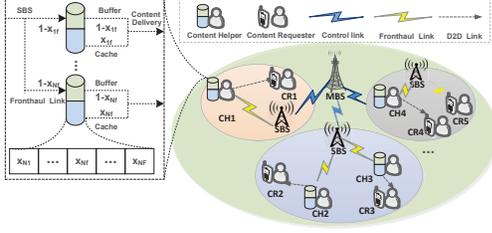[1]The dynamics can be solved in the similar way as our previous work [6].

Fig. 1: Framework for Cache- and Buffer-Enabled Social IIoT

Fig. 2: Tripartite graph for entire edge caching scheme

initial announcement sending stage and the mitigation factor to suppress the co-channel interference, so that we need not focus on the complex dynamic process of the interference itself. Similar as [11], we assume that all the SBSs, CHs, and CRs share the same bandwidth W, while the cellular links and D2D links in each cell are allocated with orthogonal bandwidth with equal size so that the interference is simplified. For CH h_i served by the SBS S_k, the transmission rate for CR r_j is given by $R_{i,j} = B_k \log_2(1 + \frac{p_{i,j}h_{i,j}}{\sum_{m \in G_i \setminus i} \epsilon p_{m,j}h_{m,j} + \sigma^2})$, where ϵ denotes the mitigation factor, G_i denotes the users sharing the same sub-channel with h_i, $B_k = W/Q_k$ is the normalised bandwidth allocated to each served user by S_k, Q_k denotes the quota that one SBS can serve different transmission mode, $h_{i,j} = K\delta_{i,j}\xi_{i,j}d_{i,j}^{-\eta}$ presents the channel gain, where K denotes the tunable parameter, the fast fading gain $\delta_{i,j}$ obeys the exponential distribution, the slowing fading gain $\xi_{i,j}$ obeys the log-normal distribution, $d_{i,j}$ is the distance between h_i and r_j, η represents the path loss exponent, and σ_n^2 is the noise variance. The content delivery rate from S_k to h_i, termed as cellular link, is formulated as $R_{k,i} = B_k \log_2(1 + \frac{p_k h_{k,i}}{\sum_{m \in G_k \setminus k} \epsilon p_{m,i}h_{m,i} + \sigma^2})$, where G_k denotes the users sharing the same sub-channel with SBS k.

As mentioned before, when the request proportion of file f in h_i is complete, i.e., $x_{if} = 1$, CH can directly deliver it to CR . When $x_{if} < 1$, the h_i served by S_k completes the content delivery procedure in two stages. In the first stage, the CR r_j downloads pre-cached $x_{if}L_f$ bits from h_i. In the second stage, h_i fetches the remained $(1 - x_{if})L_f$ bits from S_k, and then delivers these $(1 - x_{if})L_f$ bits to the CR i. In general, highest QoE levels may cause a significant amount of energy consumption in IoT scenarios [23]. Hence, it is more meaningful to jointly achieve the satisfactory QoE performance and energy saving. Inspired by this, we define the energy consumption as the cost, and minimize the cost in both cache placement and content delivery process[2]. The energy consumption for CH i served by SBS k delivering file

f to CR j can be formulated as

$$c_{ij}^k = \underbrace{\frac{\alpha P_{ij}x_{if}L_f}{R_{ij}}}_{\text{Transmission cost in the first stage}} + \underbrace{\frac{\alpha P_{ki}(1 - x_{if})L_f}{R_{ki}}}_{\text{Transmission cost in the second stage}}$$
$$+ \underbrace{\frac{\beta P_b D(1 - x_{if})L_f}{C_i - \sum_{f \in F} x_{if}L_f}}_{\text{Buffer cost in the second stage}} + \underbrace{\frac{\alpha P_{ij}(1 - x_{if})L_f}{R_{ij}}}_{\text{Transmission cost in the second stage}}$$

(1)

where P_b denotes the buffer power dissipation and D denotes the buffer delay [11]. α and β are the adjustment coefficient of consumed energy for transmission and buffer. For ease of exposition, we consider the static power dissipation for data generating, storing, and removing. The third term in equation (1) denotes the buffer energy consumed at h_i. Since the $\sum_{f \in F} x_{if}L_f$ fraction of capacity is used for caching, the $C_i - \sum_{f \in F} x_{if}L_f$ fraction of capacity is allocated to buffer. To deliver uncached $(1 - x_{if})L_f$ bits, the average portion of time required for buffer is $\frac{(1 - x_{if})L_f}{C_i - \sum_{f \in F} x_{if}L_f}$. When the storage capacity of the h_i is full, i.e., $C_i - \sum_{f \in F} x_{if}L_f = 0$, the h_i cannot fetch uncached files since it does not have extra capacity for buffering. Based on Littles law [24], the average delay for buffering file f (the period that a data packet is stored in the buffer capacity), is denoted by $\frac{D(1 - x_{if})L_f}{C_i - \sum_{f \in F} x_{if}L_f}$, which means that small buffer leads to long queue, and thus causes large delay time. Considering the effect of popularity distribution, the average cost of h_i can be given by $c_i^{avg} = \sum_{j \in CR} \sum_{f \in F} q_f c_{ij}$.

B. Social Domain

Different from the static social attributes in [2], we consider both static and dynamic social attributes involving not only their owners' inter-human social relations, but also the inter-device social relations, which can be categorized into three types: (i) co-work object relationship, which measures the similarity in task cooperation or application; (ii) social object relationship, which weights the strength of social interaction, and (iii) social reputation, which evaluates the contribution in historical task execution process. For simplicity, we use Γ_{ij} to denotes the social attributes except for social reputation.

The reward can provide incentives for CHs to accept the content sharing request. Herein, the reward function is influenced by social reputation and user evaluation. By analogy to the QoE evaluation methods in [25], we adopt the Mean Opinion Score (MOS) method to reflect the satisfaction for

---

[2]Generally, the energy consumed by content delivery is several orders of magnitudes higher than that of the context exchange, where the latter can be ignored to reduce the computation complexity.
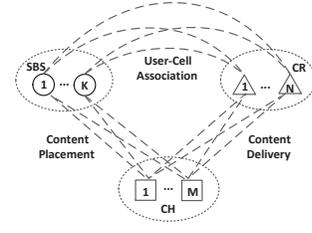
service quality. We assume that CRs can evaluate service quality and give marks to the CHs on a scale of one to five, which indicate Bad, Not so bad, Acceptable, Good, and Perfect for users QoE, respectively. $E = \{1, 2, 3, 4, 5\}$ represents the set of evaluation vale. In this paper, we assume that the CH i's social reputation SR_i before finishing the service is reflected by the service quality evaluation. Similar to [4], we consider two case for the reputation increment based on users' satisfactory level, which can be given by

$$\triangle SR_{ij} = \begin{cases} \frac{MOS(R_{ij})}{sum(E)} SR_i, & \text{if} \quad MOS(R_{ij}) > 1 \\ -\frac{n_i^1}{sum(E)} SR_i & \text{others.} \end{cases} \quad (2)$$

where $sum(E)$ denotes the sum of evaluation vale, n_i^1 is the number of times that h_i provide bad service quality. This update strategy can encourage CHs to provide better service quality. The social reputation SR_i' after finishing the service can be represented by $\min\{1, SR_i + \triangle SR_{ij}\}$, which means that the social reputation $SR_i \in [0, 1]$.

In order to trade off both the QoE level and energy consumption, The reward function is determined by the social reputation increment, the service quality evaluation from CRs to be served, and the transmission cost, which is defined as $rw_{ij} = \varepsilon \triangle SR_{ij} - \zeta c_{ij}^k$, where ε and ζ denotes the price coefficient, which can transform reputation increment energy consumption cost into a unified unit. On the one hand, the CHs will provide better QoE to obtain more reputation increment. On the other hand, the CHs will minimize the energy cost as much as possible. Based on above descriptions and assumptions, we formulate the joint cache placement and content delivery problem to maximize the rewards of CHs while guaranteeing the QoE of CRs, which can be given by

$$\max_{\omega, x, \nu} \sum_{i=1}^{N} \sum_{j=1}^{M} \sum_{k=1}^{K} \varepsilon \omega_{i,j} \triangle SR_{ij} - \zeta \omega_{i,j} \nu_{i,k} c_{ij}^k \quad (3a)$$

$$s.t. \quad C1 : rw_{ij} \geq 0; \quad (3b)$$

$$C2 : \quad \sum_{i=1}^{N} \omega_{i,j} \leqslant 1, \sum_{j=1}^{M} \omega_{i,j} \leqslant 1; \quad (3c)$$

$$C3 : \quad \sum_{i=1}^{N} \nu_{i,k} \leqslant Q_k, \sum_{k=1}^{K} \nu_{i,k} \leqslant 1; \quad (3d)$$

$$C4 : \quad \sum_{f \in F} x_{if} L_f \leq C_i, 0 \leq x_{if} \leq 1. \quad (3e)$$

where ω and ν are binary variables to determine the CH-CR pairing and user-cell association. $C1$ can guarantee the feasibility of incentive mechanism. $C2$ indicates that each CH can serve at most one CR and each CR can only download desired content from at most one CH per time. $C3$ indicates that one CH can be associated to at most one SBS and one SBS can serve at most Q_k CHs. $C4$ make sure that the sum size of cached files cannot beyond the available storage capacity and the proportion of each cached file cannot beyond the file size.

Note that problem (3) is determined by two binary variables ω and ν as well as the continuous variable x, which are related to the content delivery problem and cache placement problem, indicating that we should concurrently solve both the combinatorial optimization and linear optimization problems.

Further more, it can be observed that there exists different relationships between both two of SBS, CH, and CR, which beyonds the scope of bipartite graph. Hence, we depict the complex relationships as the tripartite graph as shown in Fig. 2 and solve it in the next section.

III. JOINT CACHE PLACEMENT AND CONTENT DELIVERY

In this section, considering that the popular contents should be fetched at CHs before content delivery, we first propose a proactive cache placement scheme to minimize the average cost ahead of time. We prove that the average cost minimization problem is not influenced by the variations in content delivery process, i.e., providing service for which CR. Hence, we can decouple the optimization problem (3) into a cache placement sub-problem and a content delivery sub-problem. After solving the first sub-problem, we further propose a ternary stable matching algorithm to solve the user pairing sub-problem in content delivery process.

A. Optimal Cache Placement Scheme

The proactive cache placement targets at potentially minimizing the average cost in the later content delivery procedure. Based on equation (1), we can observed that if the requested file is cached at this CH ahead of time, the cost will be reduced with the less buffer cost. Intuitively, the optimal cache placement scheme is determined by the popularity distribution in a certain time since those popular ones are more possibly requested by more CRs. Note that the combination of the first term and forth term in equation (1) can be represented as $\frac{\alpha P_{ij} L_f}{R_{ij}}$, which is not determined by the variable x and thus can be ignored in cache placement scheme. Meanwhile, considering that the CH has determined to fetch cached contents from the SBS k, which means that R_{ik} is given. Considering that the each CH will perform their own cache placement scheme in a distributed manner, the processes are independent of each other [11]. Hence, average cost minimization problem has a decoupled objective function and decoupled constraints. We decouple this problem into N subproblems, and subproblem i for cache placement of in h_i is formulated as

$$\min_{x_i} \sum_{f \in F} q_f \left(\frac{\alpha P_{ki} (1 - x_{if}) L_f}{R_{ki}} + \frac{\beta P_b D (1 - x_{if}) L_f}{C_i - \sum_{f \in F} x_{if} L_f} \right) \quad (4a)$$

$$s.t. \quad C1 : \sum_{f \in F} x_{if} L_f \leq C_i; \quad (4b)$$

$$C2 : \quad 0 \leq x_{if} \leq 1, \forall f \in F. \quad (4c)$$

where $x_i = [x_{i1}, ..., x_{iF}]^T$. Let $f(y, z) = \frac{y}{C_i - L_f + y - z}$, where $y = L_f(1 - x_{if})$ and $z = \sum_{l \in F \backslash f} s_{il} L_l$. Problem (4) can be proved to be non-convex due to the Hessian matrix $\bigtriangledown^2 f(y, z)$ is not positive semi-definite.

Our proposed cache placement we can still obtain a closed form for each CH despite the non-convexity of problem (4). Theorem 1 reveals the special structure of optimal solution, which demonstrates the existence of the finite solution space.

Theorem 1: For the optimal cache placement scheme x_i^* of CH i, at most one file f has the probability form with

the range $x_{if}^* \in (0,1)$, and the other file l satisfies $x_{il} = 0$ or $1, \forall l \in F \setminus f$. Beyond that, x_i^* satisfies $x_{i1}^* \geq x_{i2}^* \geq, ..., \geq x_{iF}^*$.

The Theorem 1 can simply be proved by lagrangian multiplier method. Due to the page limit, we omit the proof here. Theorem 1 indicates that there must exist a finite solution space for the optimal solution. Given all possibly optimal solutions, Theorem 2 denotes the optimal scheme.

Theorem 2: The optimal x_i^* can be chosen from one of the following $F_{i1} - F_{i2} + 1$ possibly solution space: $(1_{f-1}, x_{if}^*, 0_{F-f}), f = F_{i2}, ..., F_{i1}$.

$$F_{i1} = \begin{cases} \min_{f \in F, \sum_{l=1}^{f} L_l = C_i} f, & \text{if } \exists f \& \sum_{l=1}^{f} L_l = C_i \\ \min_{f \in F, \sum_{l=1}^{f} L_l > C_i} f, & \text{Otherwise.} \end{cases} \quad (5a)$$

$$F_{i2} = \max_{f \in F, C_i \geq \sum_{l=1}^{f} L_l + \sum_{l=f+1}^{F} \frac{q_l L_l}{q_f}} f \quad (5b)$$

Where F_{i1} represents all the potential solutions of problem (5) and F_{i2} represents the solutions to be reduced. Let $C_i^* = \sum_{l=1}^{f} L_l + \sum_{l=f+1}^{F} \frac{q_l L_l}{q_f}$, the candidate solutions of x_{if}^* can be given as follows:

$$x_{if}^* = \begin{cases} x_{if}^{max} & \text{if } C_i \geq C_i^* \\ \arg\min_{x_{if} \in \{x_{if}(1), x_{if}^{max}\}} g(x_{if}) & \text{if } C_i < C_i^* \\ \arg\min_{x_{if} \in \{0, x_{if}^{max}\}} g(x_{if}) & \text{if } C_i < C_i^* \text{ and} \\ \qquad\qquad\qquad\qquad\qquad x_{if}(1) \leq 0 \\ x_{if}^{max} & \text{if } C_i < C_i^* \text{ and } x_{if}(1) \geq x_{if}^{max} \end{cases}$$

$$(6)$$

Proof: Please refer to Appendix A.

Theorem 2 indicates that the optimal solution is one of the $F_{i1} - F_{i2} + 1$ potential solutions.

B. Distributed Content Delivery Scheme

Based on the aforementioned discussions, the entire caching scheme can be depicted as a tripartite graph as shown in Fig. 2. Binary matching in bipartite graphs and its extensions have been well studied in [19]. Different from the bipartite graph, a k-partite graph can be defined as a graph G whose nodes can be partitioned into k disjoint sets G_i, $i = 1, 2, ..., k$, i.e., k genders. Hence, no two vertices within the same set are adjacent, which indicates that each member in a gender only has a preference list for all members in other $k - 1$ genders. This graph is reduced to a bipartite graph when $k = 2$, and a tripartite graph when $k = 3$. To make this model more applicable to our content sharing scheme, the k-ary matching in this paper is defined as a set of n k-tuples, where each tuple has one member from k disjoint sets and each member belongs to at least one tuple. Based on above definitions, we can naturally formulate the entire content sharing scheme as the ternary matching since every two sets among CHs, CRs, and SBSs have relationships (vertices) with each other. The reason is that the relationship content placement between SBS and CH means that each selected CH will pre-cache the popular contents or act as buffer relay, the relationship content delivery between CH and CR implies the D2D pairing, and the relationship user-cell association between SBS and CR means that the CR can download the desired contents from SBS if the D2D link fails, to achieve the better QoE.

Definition 1: Given a set of triples, a triple is called a blocking family if each member in this triple strictly prefers each of the members in other triple to the corresponding partners the current triple. A stable ternary matching means a matching without blocking family.

In this paper, the preference profile is defined for members of each individual gender, which can be given by $U_i(j) = rw_{ij}$, $U_i(k) = R_{ki}$, $U_j(i) = (SR_i + \Gamma_{ij})R_{ij}$, and $U_k(i) = C_i \cdot SR_i$, where C_i denotes the storage capacity. Since we will prove that the $k - 1$ round matching is enough for stability, the $(k-1)^2$ number of preference values is enough. Without loss of generality, we write $h_i \succ_{r_j} h_{i'}$ to represent that r_j prefers h_i to $h_{i'}$, which is defined as $h_i \succ_{r_j} h_{i'} \Leftrightarrow U_j(i) > U_j(i')$, where $\succ$ denotes a binary preference relation. Thus, r_j's preference list over h_i is ranked by the preference value $U_j(i)$ in a descending order.

We can observe from Fig. 2 that the ternary stable matching consists of two different type of matching game. We formulate the user-cell association and CH selection problems as a many-to-one matching game, and the CH-CR pairing problem as an one-to-one matching game. To find a stable ternary matching, we will prove that 2-round binary matching among three genders is enough. Besides, we will prove that more than 2-round bindings may not always exist with strengthened family tie and any less than 2-round bindings with loosened family tie may cause instability. In a word, 2-round matching is enough to guarantee the stability.

Our proposed approach to find a stable ternary matching is based on the Gale-Shapley (GS) algorithm in [19] and MSU algorithm in [6] for tripartite graphs, by iteratively and pairwisely binding all disjoint sets in a binary matching through a spanning tree. The GS algorithm consists of a number of rounds of proposing, accepting/rejecting to find a stable ternary matching. In each round, the unaccepted CRs requests to the most preferred CHs to whom he has not yet requested, and then each CHs accepted the most preferred suitor while rejects the rest. The iteration terminates until there exists any CH who still has vacancy. Similarly, the difference between GS algorithm and MSU algorithm is that each SBS can accept more than one proposer (CH) within a fixed quota. In particular, one application of the GS algorithm on CH and CR is denoted as $GS(i, j)$, to find the one-to-one stable binary matching. And one application of the MSU algorithm on CH and SBS is defined as $MSU(i, k)$ aiming at obtaining a many-to-one stable matching with quota. Relation $i - j$ presents the binding of CH and CR. Two rounds of the approach are implemented, thereby a spanning binding tree T is constructed in this way. $V(T)$, $E(T)$, P represents the node set, edge set of T, and pairwisely bindings, respectively in Algorithm 1.

Theorem 3: The DTSM algorithm generates a stable ternary matching.

Proof: Assuming that the final matching Ψ contains at least one blocking family, there must exist two tuples: (S_k, h_i, r_j) and $(S_k', h_{i'}, r_{j'})$, $\exists m \in (S_k, h_i, r_j)$ and $\exists m' \in (S_k', h_{i'}, r_{j'})$, satisfying $m' \succ_m \Psi(m)$ and $m \succ_{m'} \Psi(m')$. This condition contradicts the fact that both GS algorithm and MSU algorithm generate binary stable matchings. Theorem 3 also demonstrates that $k - 1$ rounds of binding is tight enough to

Algorithm 1 Distributed Ternary Stable Matching Algorithm (DTSM)

1: Establish the preference list PL^k for SBSs, PL^i for CHs, and PL^j for CRs. Let $V(T) = \{i\}$ and $E(T) = P = \{\}$
2: **while** T is not a spanning tree **do**
3:    $V(T) = V(T) \cup j \cup k$, $E(T) = E(T) \cup (i,j) \cup (i,k)$;
4:    Let CR as requester in the GS algorithm and CH as proposer in the MSU algorithm, $P = P \cup GS(i,j) \cup MSU(i,k)$;
5: **end while**
6: Obtain the ternary stable matching Ψ derived from P.

---

make sure the stability. The stability with loop tie may not exist so that more than $k - 1$ rounds is not needed, and less than $k - 1$ round will cause instability because there may exist a blocking family if any member without any binding. As for our model, although we eliminate the tie between SBSs and CRs, the binding between CRs and CHs can guarantee that the user-cell association is stable.

IV. INCONSPICUOUS MANIPULATION AND ANALYSIS

In this section, we will investigate an interesting and challenging topic: "How to inconspicuously manipulate the matching result in above section to further benefit more CHs?". In detail, suppose a CH can manipulate by permuting it's true preference list, can we obtain a CH-optimal one-to-one matching while preserving stability? Besides optimality and stability, the manipulator may also want to get rid of the suspicion of misreporting. In practice, it is reasonable to expect that sometimes the scheme designer glances through the true preference lists of all members and ignores some minor changes, say from past resulting data of the algorithm. In this case, misreporting can be easily detected if the misreported lists looks significantly different from the estimate. However, the lack of necessary theoretical proof cannot prove this standpoint since we cannot know why this strategy works. Inspired by this work, we prove for the first time that the CH-optimal matching can be obtained with the existence of domino effect and then we can derive an equivalent inconspicuous manipulation for optimal manipulation leading to the same result from the true list by promoting only one CR in this CH's preference list. We first give some notations. The preference profile $\succ = \{\succ_i, \succ_{-i}, \succ_j, \succ_{-j}\}$ consisting of the preference lists of all CHs and CRs, where $\succ_{-i}$ represents the preference lists of all CHs except CH i. As defined before, we use $j \succ_i j'$ to denote that CH i prefers CR j' to j, and $j \succeq_i j'$ to denote that CH i prefers CR j' at least as j. The matching result obtained in GS algorithm is $\Omega = GS(\succ)$. Ω admits a blocking pair with regard to $\succ$ if there exists a pair (i,j) satisfying that $j \succ_i \Omega(i)$ and $i \succ_j \Omega(j)$. Ω is stable if containing none blocking pair with respect to $\succ$. $G_\succ$ represents the set of all stable matchings with respect to $\succ$. For any $\Omega \in G_\succ$, Ω is CR-optimal if $\Omega(j) \succeq_j \Omega'(j)$ and for all CRs and $\Omega'(i) \succeq_i \Omega(i)$ and for all CHs, which conforms to the concept of CH-optimal matching. Next, We let $Suitor(CH, \succ)$ denote the CRs requesting to CH i during

the run of GS algorithm with preference lists $\succ$. In addition, we let $Suitor(CH, \succ, j)$ denote the j-th preferable CR of CH in $Suitor(CH, \succ)$. Naturally, $Suitor(CH, \succ, 1) = \Omega(i)$ for $\Omega = GS(\succ)$.

Without loss of generality, we can manipulate the stable matching by an agent i, i.e., manipulator, if there exists $\succ$ and $\succ'$ differing only in the preferences of i, such that $\Omega'(i) \succ_i \Omega(i)$, where Ω and Ω' are the matching results before and after the manipulation, respectively. Similarly, we use $\succ' = \{\succ'_i, \succ_{-i}\}$ and $\Omega' = GS(\succ')$ to denote the manipulated preference and the matching result, respectively. The manipulation $\succ'_i$ with respect to $\succ$ is stability-preserving if $\Omega' \in G_\succ$. Given a preference profile $\succ$, an optimal manipulation of GS algorithm by i with regard to $\succ$ is denoted as $\succ'_i$ that satisfies $\Omega'(i) \succ_i \Omega(i)$, and $\Omega'(i) \succeq_i \Omega''(i)$ for any other preference list $\succ''$ where $\Omega'' = GS(\succ'')$. We call $\succ''_i$ an inconspicuous equivalent of a manipulation $\succ'_i$ if $\succ''_i$ can be derived from $\succ_i$ by permuting only one position and $\Omega''(i) = \Omega'(i)$.

Theorem 4: Suppose that $\succ'_i$ is an optimal manipulation with regard to $\succ$ for CH i and $\Omega' = GS(\succ')$. $\succ'_i$ is stability-preserving because $\Omega' \in G_\succ$.

Proof: Before giving the proof, we can observe that those CRs proposing to CH i in the procedure $GS(\succ)$ will also propose to i in the procedure $GS(\succ')$,i.e., $j \in Suitor(i, \succ) \Rightarrow j \in Suitor(i, \succ')$. Because these CRs will be matched with the less preferred or present CH in Ω' and thus they will still propose to the CHs in $GS(\succ)$.

Then we prove Theorem 4 by contradiction. Assuming that $\Omega' \notin G_\succ$, there must exist a blocking pair (i', j') that satisfying $j' \succ_{i'} \Omega'(i')$ and $i' \succ_{j'} \Omega'(j')$. We first consider the case $i' = i$. Given $\Omega' \in G_{\succ'}$, we have either $\Omega'(j') \succ'_{j'} i'$ or $\Omega'(i') \succ'_{i'} j'$. If $i' \neq i$, we have $\succ'_{i'} = \succ_{i'}$ and $\succ'_{j'} = \succ_{j'}$. We further obtain that $\Omega'(j') \succ_{j'} i'$ or $\Omega'(i') \succ_{i'} j'$, which contradicts to the precondition of blocking pair. Hence, we have $i' = i$. Considering that the $\succ''_i$ is derived from $\succ'_i$ by moving j' to the first position, we have $\succ'' = \{\succ''_i, \succ_{-i}\}$ and $\Omega'' = GS(\succ'')$. Given $i' = i$, we have $j' \succ_i \Omega'(i)$ or $i \succ_{j'} \Omega'(j')$. Since $\succ'_{j'} = \succ_{j'}$, we also have $i \succ'_{j'} \Omega'(j')$. As a result, during the run of $GS(\succ')$, j' must propose to and be rejected by i before matched with $\Omega'(j')$ due to the transformation that $j' \in Suitor(i, \succ') \Rightarrow j' \in Suitor(i, \succ'')$. Based on the precondition that j' is in the first position of $\succ''_i$, we have $\Omega''(i) = j'$, which contradicts to the optimality of manipulation $\succ'_i$ due to the fact that $j' \succ_i \Omega'(i)$.

Lemma 1: If CH i swaps the position $j \in Suitor(i, \succ, 1)$ and $j' \in Suitor(i, \succ, 2)$. There may exists a circle in which all involved CHs rotation-swap their matched CRs and then all involved CHs get preferable CRs.

To prove Lemma 1, we analyze the Example 1 in Table I. Note that when applying GS algorithm with the true preference list, h_1 receives requesting only from r_4 and r_5. Therefore, any manipulation by h_1 must involve the order swap of r_4 and r_5. By rejecting r_4 and accepting r_5, r_4 is forced to request to h_3, and then replacing the r_3 as the current partner of h_3. Therefore, this replacement force r_3 to request to h_1. Now h_1 has already been matched to a preferred partner r_3 than original partner r_4. However, h_1 can still get better by pretending to prefer r_5 to r_3 . This manipulation forces r_3 to

request to h_4 and thus replacing r_2, who in turn is forced to request to h_1, thereby giving h_1 a better match.

Example 1 points out that swapping the position of first and the second-best requesters can result in at most one new requester, which is more preferred by the manipulator. In the result matching of optimal manipulation, at least two CHs (h_1, h_3, and h_4 in example 1) will rotationally swap their matched CRs. h_1 is matched with h_4's original partner r_2, h_4 is matched with h_3's original partner r_3, and h_3 get h_1's original partner r_4. If the manipulator is involved in this loop, the manipulator will get desired CR by swapping the position of the first and second-best requesters. Otherwise, the manipulator will get less preferred CR. Based on above discussions, we can iteratively utilize the Domino effect until obtaining an optimal manipulation. Note that it is still possible

TABLE I: Example 1(The existence of Domino Effect)

True preferences of CRs	True preferences of CHs
$r_1 : h_2\ h_1\ h_3\ h_4\ h_5$	$h_1 : r_1\ r_2\ r_3\ \underline{r_4}\ r_5$
$r_2 : h_4\ h_1\ h_5\ h_3\ h_2$	$h_2 : r_3\ r_2\ r_4\ r_5\ \underline{r_1}$
$r_3 : h_3\ h_1\ h_4\ h_5\ h_2$	$h_3 : r_4\ r_2\ r_1\ r_5\ \underline{r_3}$
$r_4 : h_1\ h_3\ h_4\ h_5\ h_2$	$h_4 : r_5\ r_1\ r_4\ r_3\ \underline{r_2}$
$r_5 : h_1\ h_5\ h_4\ h_3\ h_2$	$h_5 : r_1\ r_2\ r_3\ r_4\ \underline{r_5}$
Optimal + Non-Inconspicuous	**Optimal + Inconspicuous**
$h_1 : \underline{r_2}\ r_5\ r_4\ r_3\ r_1$	$h_1 : r_1\ \underline{r_2}\ r_5\ r_3\ r_4$
$h_2 : r_3\ r_2\ r_4\ r_5\ \underline{r_1}$	$h_2 : r_3\ r_2\ r_4\ r_5\ \underline{r_1}$
$h_3 : \underline{r_4}\ r_2\ r_1\ r_5\ r_3$	$h_3 : \underline{r_4}\ r_2\ r_1\ r_5\ r_3$
$h_4 : \underline{r_5}\ r_1\ r_4\ \underline{r_3}\ r_2$	$h_4 : r_5\ r_1\ r_4\ \underline{r_3}\ r_2$
$h_5 : r_1\ r_2\ r_3\ \underline{r_4}\ \underline{r_5}$	$h_5 : r_1\ r_2\ r_3\ \underline{r_4}\ \underline{r_5}$

that the list misreported in optimal manipulation might look very different from the true list, which motives us to find a inconspicuous manipulation. Theorem 5 demonstrates that the optimal manipulation must has an inconspicuous equivalent leading to the same matching result while differing from the true list in only one position.

Theorem 5: Assume that $\succ_i'$ is an optimal manipulation with regard to $\succ$ for i and $\succ_i'$ is another preference list derived from $\succ_i$ by moving the $j \in Suitor(i, \succ', 2)$ to the position right after $j' \in Suitor(i, \succ', 1)$ while making no other changes. As a result, $\succ_i'$ is the inconspicuous equivalent of $\succ_i'$, i.e., $\Omega''(i) = \Omega'(i)$.

Proof: Theorem 5 can be easily proved by Lemma 1. Moving the $Suitor(i, \succ', 2)$ to the position right after $Suitor(i, \succ', 1)$ can make sure that the $Suitor(i, \succ', 1)$ will request to i and i will reject the requesters in the position between $Suitor(i, \succ', 1)$ and $Suitor(i, \succ', 2)$ with respect to the true preference list. Finally, we have $\Omega''(i) = \Omega'(i)$.

Based on the discussion, the entire content sharing scheme can be summarized in Algorithm 2.

Since Theorem 4 prove that the optimal manipulation with respect to the true preference list and thus the Λ is stability-preserving, we only provide analysis on complexity properties.

Complexity Analysis: The computational complexity of the proposed JCPD algorithm is formulated as $\mathcal{O}(\log(d) \sum_{i \in CH}(F_{i1} - F_{i2}) + M^2N^2 + \max\{MN, KN\})$ in the worst case. Considering the combinatorial nature, the computational complexity of each sub-algorithm included in JCPD algorithm will be analysed in turn. For the optimal cache placement scheme in first stage, the optimal x_i^* is one of the $F_{i1} - F_{i2}$ potential solutions. According to [26], the complexity of obtaining k-th root for each potential solution is $\mathcal{O}((2\log(k))\log(d))$ using the Newtons

---

Algorithm 2 Joint Content Placement and Delivery Algorithm (JCPD)

1: ***Step 1: CH Selection and Cache Placement:***
2: CHs exchanges context information with SBSs, and obtain the stable matching using MSU algorithm. Given the matching result, each CH will pre-cache a subset of all files based on the optimal cache placement scheme.
3: ***Step 2: Distributed Content Delivery:***
4: CHs exchanges context information with CRs, and then the CRs send their request based on the preference profile. Each CH determines whether accepting the request based on the potential reward increment and cost.
5: The selected CH will first perform ***manipulation*** based on Lemma 2 and then inconspicuous manipulate the true preference list based on the optimal manipulation.
6: Repeat ***manipulation*** until there exists no Domino effect in current matching state Λ.

---

method, where d represents the number of precision bits desired. In sum, the total complexity of the optimal cache placement scheme is $\mathcal{O}(\log(d) \sum_{h_i \in CH}(F_{i1} - F_{i2}))$. Based on [6], the computational complexity of GS algorithm and MSU algorithm can be denoted as $\mathcal{O}(MN)$ and $\mathcal{O}(NK)$, respectively. By parallel processing, the computational complexity of DTSM is $\mathcal{O}(\max\{MN, KN\})$. The computational complexity of optimal manipulation is $\mathcal{O}(\mathcal{T}MN)$, where $\mathcal{T}$ is the iteration number of obtaining the optimal manipulation.

We can also observe that the iteration number is upper bounded by $N(M - 1)/2$. Considering the worst case that the original partner is in the end position and can only move one position with respect to the true preference list, the manipulator can move at most M-1 position. One manipulation can benefit at least two users per time. Hence, the computational complexity is $\mathcal{O}(N^2M^2)$ in the worst case. Actually, due to the lattice of preference profile, the worst case will never happen, and the iteration number is smaller than M in practice. Compared to the centralized expanded KM (EHM) algorithm in [14] with the computational complexity $\mathcal{O}(\max\{N, M, K\}^3(1 + \min\{A_{\max\{N,M\}}^{\min\{N,M\}}, A_{\max\{N,K\}}^{\min\{N,K\}}\}))$ in the worst case, which can obtain a maximal-weight unstable three dimensional matching, our distributed scheme can achieve the near-optimal performance with lower computational complexity and stability guarantee.

V. SIMULATION RESULTS AND PERFORMANCE ANALYSIS

In this section, we conduct numerical simulations generated on realistic social dataset to demonstrate the validity of the theoretical analysis and the effectiveness of our proposed algorithm. Based on the realistic CRAWDAD dataset [27], we can extract social attributes for 72 participants based on the method in [6]. We comprehensively evaluate our proposed algorithm, abbreviated as JCPD in terms of reward, convergence, average cost, and user satisfaction performance. The main simulation parameters are listed in Table II. We set the parameter according to [11], [22], which is applicable
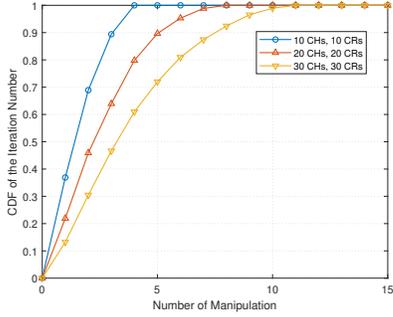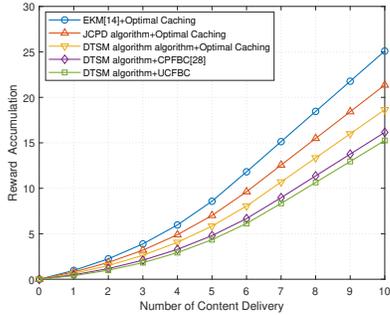
Fig. 3: Distribution of iterations

Fig. 5: Average number of users in different MOS

Fig. 4: Average reward accumulation

Fig. 6: Average energy consumption

for wireless communication of most heterogeneous cellular network. Furthermore, we use four benchmark algorithms to evaluate our proposed algorithm. The first benchmark algorithm is EKM in [14], which can achieve the maximal weight in content delivery process. The second benchmark is "DTSM" algorithm in Algorithm 1. In the third benchmark, we adopt the cache placement algorithm with fixed buffer capacity (CPFBC) in [28], in which the cache capacity is set equal to the buffer capacity. For the final benchmark, we adopt uniform caching with fixed buffer capacity, abbreviated as UCFBC. In this paper, we only investigate the cache placement scheme for MUs, and the buffer of SBSs is not considered for simplicity. Each SBS will cache part of the complete contents based on the Theorem 1 without considering the buffer capacity.

TABLE II: Simulation Parameters

Parameters	Value
Macro cell radius and Small Cell radius	1000m,300m
Number of SBSs	5
Number of CHs and CRs	72 [27]
The available storage space of of CHs and SBSs	100-500Mbits, 1000-2000Mbits
Buffer delay and power dissipation	6s, 125mW [11]
price coefficient ε, ζ	20, 0.1
Adjustment coefficient α and β	1, 1
Mitigation factor ϵ	60dB [22]
Zipf parameter γ, Number of files F	0.9, 30
Size of each file L	10-100Mbits
Transmission power of SBS and CH	500mW, 125mW
Bitrate θ_1 and θ_4 for HD video stream	1.5Mbps, 4Mbps
Noise power spectral density σ^2	-144dbm/Hz
Bandwidth W and quota Q	10MHz, 12
Multi-path fading δ, Shadowing ξ	1, 8db
Pass-loss exponent, Pass-loss constant K	4, 10^{-2}

Firstly, in order to demonstrate the convergence behavior of our proposed algorithm, we investigate the cumulative distribution function (CDF) versus the number of manipulation required. In Fig. 3, we plot the CDF of manipulation number when reaching to the stable matching state, thereby demonstrating the validity of the complexity analysis. From
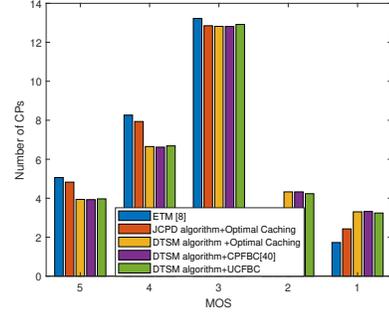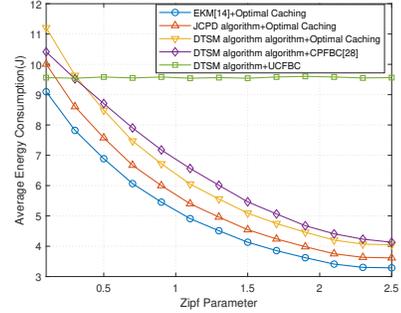
this figure, it can be observed that our proposed algorithm converges faster with the increasing number of CHs, due to the fact that finding the domino effect is easier. In addition, Fig. 3 further reveals that the needed convergence number is relatively small with respect to the user number. Fig. 4 shows the reward increment with the increasing number of content delivery. We assume that half of the MUs request contents in each content delivery process, and the request probability follows the Zipf distribution. At the very beginning, due to the relatively low initial reputation level, the reward gap between any two schemes is not so much. The reason for the growing gap is that the schemes which can provide better QoS level and reduce the cost, will get more rewards. This simulation results show that both our proposed content delivery and cache placement schemes can always reach the near-optimal solution. More specially, we further analyse both the QoE level and energy consumption performance, respectively.

In Fig. 5, we adopt user satisfaction as a crucial metric, which is defined as the average number of CRs in different MOS. In this simulation, other cache placement schemes can achieve similar QoE level, because both the CPFBC and uniform cache scheme can provide satisfactory QoE when CRs request popular contents. In this regard, our proposed content delivery scheme can further improve the QoE level, and can always approximate the optimal performance of EKM with less complexity consumption. In order to evaluate the impact of popularity distribution on the average energy consumption, we plot the average cost varying Zipf parameter γ from 0.1 to 2.5 in Fig. 6. With the increase of γ, the popularity of top ranked contents tend to increase, and thus

pre-caching more popular contents can decrease the average energy consumption. It is worth mentioning that when γ is small, i.e., $\gamma \in [0, 0.4]$, the difference of the popularity among various contents are not obvious. In this case, both the CPFBC and uniform cache scheme can make better use of the buffer capacity, which guarantees the lower buffer cost. But our optimal cache placement scheme shows a rapidly decreasing trend with the growth of γ, because more CRs will request contents with higher popularity, which reduces the buffer cost. When the γ is very large, i.e., $\gamma \in [2, 2.5]$, the energy consumption has slowly paced down, since almost all CHs can deliver complete contents so the main cost comes from the directly transmission instead of buffering. Note that the popularity distribution has no impact on the average cost performance of uniform caching scheme, since each file is cached with same fraction.

VI. CONCLUSION

In this paper, the scheme of joint cache placement and delivery in Social IIoT has been proposed to investigate the issue of optimizing the reward of CHs while guaranteeing the QoE of CRs. Specifically, we designed a proactive cache placement scheme for the purpose of minimizing the energy consumption. In this setting, the optimization problem is to balance the QoE level and energy consumption by maximizing the reward of CHs. We conceived this scenario with framework of tripartite graph. According to this graph, we proposed a distributed algorithm to find a ternary stable matching. Last but not the least, we inconspicuously manipulated the matching with the existence of domino effect to further benefit more CHs. The simulation results generated on realistic social dataset showed that our proposed algorithms could achieve a better trade-off among energy consumption and QoE performance with relatively less complexity consumption.

APPENDIX A
PROOF OF THEOREM 2

According to Theorem 1, the optimal solution of problem (4) has the structure $(1_{f-1}, x_{if}^*, 0_{F-f})$. As a consequence, the optimal solution has F possibly solutions, $(x_{if}^*, 0_{F-1})$,..., $(1_{F-1}, x_{if}^*)$. For the f-th solution $(1_{f-1}, x_{if}^*, 0_{F-f})$, constraint $\sum_{l=1}^{f-1} L_l \leq C_i$ should be satisfied and then the optimal solution x_{if}^* can be obtained by substituting the optimal values of other $F-1$ variables into problem (4) as $\min_{x_{if}} (\frac{\alpha P_{ki}}{R_{ki}} + \frac{\beta P_b D}{C_i - \sum_{l=1}^{f-1} L_l - x_{if} L_f})(q_f(1 - x_{if})L_f + \sum_{l=f+1}^{F} L_l q_l) \overset{\triangle}{=} g(x_{if})$ with the constraint $0 \leq x_{if} \leq x_{if}^{max}$, where $x_{if}^{max} = \min\{1, \frac{C_i - \sum_{l=1}^{f-1} L_l}{L_f}\}$. Letting the first-order derivative of the objective function $g'(x_{if}) = 0$, we have the following two cases: 1) If $q_f C_i + \sum_{l=f+1}^{F} q_l L_l + \sum_{l=1}^{f} q_f L_l \leq 0$, the right term of this equation is always non-positive and thus $g'(x_{if}) \leq 0$. The objective function $g(x_{if})$ monotonically decreases with x_{if}, and the optimal $x_{if}^* = x_{if}^{max}$. 2) If $q_f C_i + \sum_{l=f+1}^{F} q_l L_l + \sum_{l=1}^{f} q_f L_l > 0$, two different roots can be represented as $x_{if}(1) = \frac{(C_i - \sum_{l=1}^{f-1} L_l)\sqrt{q_f} - \sqrt{\rho}}{L_f \sqrt{q_f}}$ and $x_{if}(2) = \frac{(C_i + \sum_{l=1}^{f-1} L_l)\sqrt{q_f} + \sqrt{\rho}}{L_f \sqrt{q_f}}$, where $\rho = R_{ki} \beta P_b D(-q_f C_i + \sum_{l=f+1}^{F} q_l L_l + \sum_{l=1}^{f} q_f L_l)/$

αP_{ki}. Since the objective function $g(x_{if})$ decreases with x_{if} when $x_{if} < x_{if}(1)$ and $x_{if} > x_{if}(2)$ while increasing with x_{if} when $x_{if}(1) \leq x_{if} \leq x_{if}(2)$. If $0 \leq x_{if} \leq x_{if}^{max}$, we have the optimal $x_{if}^* = \arg \min_{x_{if} \in \{x_{if}(1), x_{if}^{max}\}} g(x_{if})$. If $x_{if}(1) \leq 0$, $x_{if}^* = \arg \min_{x_{if} \in \{0, x_{if}^{max}\}} g(x_{if})$. If $x_{if}(1) \geq x_{if}^{max}$, $x_{if}^* = x_{if}^{max}$.

REFERENCES

[1] E. Sisinni, A. Saifullah, S. Han, U. Jennehag, and M. Gidlund, "Industrial Internet of Things: Challenges, opportunities, and directions," *IEEE Transactions on Industrial Informatics*, vol. 14, pp. 4724–4734, Nov 2018.

[2] M. Nitti, L. Atzori, and I. P. Cvijikj, "Network navigability in the social Internet of Things," in *Internet of Things*, pp. 405–410, 2014.

[3] F. Jameel, S. Wyne, D. N. K. Jayakody, G. Kaddoum, and R. OKennedy, "Wireless social networks: A survey of recent advances, applications and challenges," *IEEE Access*, vol. 6, pp. 59589–59617, 2018.

[4] Z. Su, Q. Qi, Q. Xu, S. Guo, and X. Wang, "Incentive scheme for cyber physical social systems based on user behaviors," *IEEE Transactions on Emerging Topics in Computing*, pp. 1–1, 2018.

[5] D. Wu, Q. Liu, H. Wang, Q. Yang, and R. Wang, "Cache less for more: Exploiting cooperative video caching and delivery in d2d communications," *IEEE Transactions on Multimedia*, vol. 21, pp. 1788–1798, July 2019.

[6] B. Wang, Y. Sun, S. Li, and Q. Cao, "Hierarchical matching with peer effect for low-latency and high-reliable caching in social IoT," *IEEE Internet of Things Journal*, vol. 6, pp. 1193–1209, Feb 2019.

[7] B. Chen, C. Yang, and A. F. Molisch, "Cache-enabled device-to-device communications: Offloading gain and energy cost," *IEEE Transactions on Wireless Communications*, vol. 16, pp. 4519–4536, July 2017.

[8] T. Zhang, H. Fan, J. Loo, and D. Liu, "User preference aware caching deployment for device-to-device caching networks," *IEEE Systems Journal*, vol. 13, pp. 226–237, March 2019.

[9] F. Cheng, G. Gui, N. Zhao, Y. Chen, J. Tang, and H. Sari, "Uav-relaying-assisted secure transmission with caching," *IEEE Transactions on Communications*, vol. 67, pp. 3140–3153, May 2019.

[10] N. Zhao, F. Cheng, F. R. Yu, J. Tang, Y. Chen, G. Gui, and H. Sari, "Caching uav assisted secure transmission in hyper-dense networks based on interference alignment," *IEEE Transactions on Communications*, vol. 66, pp. 2281–2294, May 2018.

[11] Z. Yang, C. Pan, Y. Pan, Y. Wu, W. Xu, M. Shikh-Bahaei, and M. Chen, "Cache placement in two-tier hetnets with limited storage capacity: Cache or buffer?," *IEEE Transactions on Communications*, vol. 66, pp. 5415–5429, Nov 2018.

[12] N. Zlatanov, R. Schober, and P. Popovski, "Buffer-aided relaying with adaptive link selection," *IEEE Journal on Selected Areas in Communications*, vol. 31, pp. 1530–1542, August 2013.

[13] L. Xiang, D. W. K. Ng, T. Islam, R. Schober, V. W. S. Wong, and J. Wang, "Cross-layer optimization of fast video delivery in cache-and buffer-enabled relaying networks," *IEEE Transactions on Vehicular Technology*, vol. 66, pp. 11366–11382, Dec 2017.

[14] L. Wang, H. Wu, and Z. Han, "Wireless distributed storage in socially enabled D2D communications," *IEEE Access*, vol. 4, pp. 1971–1984, 2017.

[15] J. Wu, "Stable matching beyond bipartite graphs," in *2016 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*, pp. 480–488, May 2016.

[16] R. Vaish and D. Garg, "Manipulating gale-shapley algorithm: Preserving stability and remaining inconspicuous," in *Twenty-Sixth International Joint Conference on Artificial Intelligence*, pp. 437–443, 2017.

[17] M. S. Pini, F. Rossi, K. B. Venable, and T. Walsh, "Manipulation complexity and gender neutrality in stable marriage procedures," *Autonomous Agents and Multi-Agent Systems*, vol. 22, pp. 183–199, Jan 2011.

[18] Y. Gu, Y. Zhang, M. Pan, and Z. Han, "Matching and cheating in device to device communications underlying cellular networks," *IEEE Journal on Selected Areas in Communications*, vol. 33, pp. 2156–2166, Oct 2015.

[19] D. Gale and L. S. Shapley, "College admissions and the stability of marriage," *American Mathematical Monthly*, vol. 120, no. 69, pp. 9–15, 2013.

[20] X. Li, X. Wang, K. Li, Z. Han, and V. C. M. Leung, "Collaborative multi-tier caching in heterogeneous networks: Modeling, analysis, and design," *IEEE Transactions on Wireless Communications*, vol. 16, pp. 6926–6939, Oct 2017.

[21] N. Jindal, P. R. Panda, and S. R. Sarangi, "Reusing trace buffers to enhance cache performance," in *Design, Automation & Test in Europe Conference & Exhibition*, pp. 572–577, 2017.

[22] K. Wu, M. Jiang, and H. Tan, "D2D relay selection based on joint fuzzy and entropy theories with social similarity," *IEEE Transactions on Vehicular Technology*, vol. 67, pp. 8796–8807, Sep. 2018.

[23] H. Elhammouti, E. Sabir, M. Benjillali, L. Echabbi, and H. Tembine, "Self-organized connected objects: Rethinking qos provisioning for iot services," *IEEE Communications Magazine*, vol. 55, pp. 41–47, Sep. 2017.

[24] W. H. Kang, S. W. Lee, and B. Moon, "Flash-based extended cache for higher throughput and faster recovery," vol. 5, no. 11, pp. 1615–1626, 2012.

[25] R. C. Streijl, S. Winkler, and D. S. Hands, "Mean opinion score (mos) revisited: methods and applications, limitations and alternatives," *Multimedia Systems*, vol. 22, no. 2, pp. 213–227, 2016.

[26] S.-G. Chen and P. Hsieh, "Fast computation of the nth root," *Computers & Mathematics with Applications*, vol. 17, no. 10, pp. 1423–1427, 1989.

[27] R. I. Ciobanu and C. Dobre, "CRAWDAD dataset upb/hyccups (v. 2016-10-17)." Downloaded from https://crawdad.org/upb/hyccups/20161017, Oct. 2016.

[28] D. Liu and C. Yang, "Optimizing caching policy at base stations by exploiting user preference and spatial locality," *CoRR*, vol. abs/1710.09983, 2017.