Joint Optimization of Content Placement and Request Redirection in Mobile-CDN

Jiayi Liu* Qinghai Yang* Gwendal Simon†
*State Key Laboratory on ISN, Xidian University, China, jyliu/ qhyang@xidian.edu.cn
†Télécom Bretagne, France, gwendal.simon@telecom-bretagne.eu

***Abstract*—In a *Mobile-CDN*, Base Stations (BSs) are equipped with storages for replicating content, and they are allowed to cooperate in replying user requests through backhaul links. In this paper, we investigate the joint optimization problem of content placement and user request redirection for such a BS-based mobile CDN system. Specifically, each BS maintains a transmission queue for replying user requests issued from other BSs. Due to the limited link capacity and the dynamic network environment, the optimization problem should be jointly considered with the transmission queue states. We employ the *Stochastic optimization model* to minimize the long-term time-average transmission cost under content availability and network stability constraints. By applying the Lyapunov optimization technique, we transform the long-term problem into a set of linear programming (LP) problems, which are solved in each short time duration. Further, we propose a semi-distributed online algorithm to jointly decide content placement and user request redirection. The evaluation confirms that our solution guarantees network stability comparing to the traditional user request redirection scheme.**

I. Introduction

With the development of Network Function Virtualization (NFV) and Software Defined Network (SDN) standards, Mobile Network Operators (MNOs) are offered with the opportunity to deploy Content Delivery Network (CDN) functionality within the mobile network. For instance, Base Stations (BSs) can be equipped with storage and computing capabilities to establish the *Mobile-CDN* system to facilitate content distribution in mobile networks [1]. This integrated mobile network service is the main research direction for the upcoming 5G era. On one hand, the availability of content near the end-users improves the Quality of Experience (QoE) of mobile users, on the other hand Mobile-CDN mitigates the traffic burden on the mobile core network. In this paper, we consider such a mobile CDN system in which cooperative BSs leverage their storage and computing capabilities to replicate content and to reply mobile users' content requests.

The first proposals toward the design of cooperative Mobile-CDN [2, 3] consider the same cooperative principles as in the traditional CDN systems for wired networks. A mobile user, who is attached to a given BS, sends requests for content to this BS, which checks whether the requested content is stored in its local cache. In case of a miss, the request is redirected to another BS, from which the requested content is replied in case of a hit. Cooperative mechanisms are known to enhance the diversity of replicated content and to improve the overall hit-ratio of the system. To implement a cooperative CDN, there are two important issues to be considered: 1) the *content placement* problem decides content replication on each BS storage; 2) the *user request redirection* problem determines to which BS a missing request should be redirected.

Both problems are tightly coupled. Content placement determines content availability on each BS, and the latter affects the decision on user request redirection. The joint optimization problem of content placement and request redirection has been extensively investigated in the literature related to CDN [4, 5]. However, these works ignore the transmission capacity of the replying caches. User requests can be redirected to another cache having a high probability to be a hit for a requested content, without taking into account the link capacity and workload of the cache. Such redirection mechanism can lead to traffic congestion and unbalanced work load. In mobile networks where BSs are standard network equipments connected by constrained backhauls, avoiding network congestion and balancing traffic loads for BSs are two key requirements for designing the user request redirection mechanism of Mobile-CDNs. Moreover, mobile networks are highly dynamic. Network parameters (such as user demands distribution) can vary frequently due to user mobility. This characteristic calls for designing online algorithm which achieves the optimization goal without requiring a-priori knowledge on the statistical distribution of the network parameters.

We propose a joint content placement and user request redirection mechanism in Mobile-CDNs. The optimization goal is to minimize the overall inter-BSs transmission cost incurred by the redirection traffic. Specifically, we utilize a *Stochastic optimization model* to cope with variable network environment parameters. We aim at minimizing the long-term time-average transmission cost while at the same time at ensuring network stability to avoid BSs forwarding traffic congestions. We use the *Lyapunov optimization* framework to transform the long-term optimization problem into a set of linear programming (LP) problems, which are solved in each short time duration. Based on this method, we propose a semi-distributed online algorithm to decide content placement and request redirection without requiring any a-priori knowledge on network parameters. The online nature of the algorithm makes it suitable to be practically implemented to cope with the mobile network dynamics. Finally, simulation results confirm that our proposal guarantees network stability by comparing to the traditional user request redirection scheme.

The remainder of this paper is organized as follows. Related

works are summarized in Section II, followed by the system model description in Section III. Then, we detail the stochastic optimization formulation and its transformation into LP problems in Section IV. Our proposed semi-distributed online joint content placement and request redirection algorithm is presented in Section V. The evaluation is provided in Section VI. Finally, we conclude this work in Section VII.

II. Related Works

The MNOs have growing interest in integrating CDN functionalities into the mobile network infrastructures. In the literature, there are some works related to Mobile-CDNs. In [6], the authors motivate the deployment of CDN serving point nodes to enhance the delivery of progressive video streaming services in mobile network. They analyze the benefits of employing an Mobile CDN system for MNO on guaranteing user QoE. In [7], the authors show the benefits of deploying CDN serving points for mobile content delivery in terms of transmission cost saving. In [8], the authors integrate CDN mechanisms into mobile platforms by exploring storage capacities of mobile devices. They address the content replication problem in this scenario. In particular, leveraging the storage capacities of BSs facilitates the implementation of CDN in mobile network as the recent efforts at ETSI MEC and NFV go into this direction. In [9], the authors study the implementation of CDN mechanisms by deploying storage capacities on BSs. They show BS cooperation on caching content can enhance the mobile CDN performance in terms of enhancing hit-ratio and reducing transmission cost. However, jointly determining content placement and user redirection has not received enough attention for mobile CDN.

Distributed in-network caching is becoming increasingly important to improve user Quality of Experience for content distribution in mobile networks. To facilitate the distributed storage of content, network coding is a useful tool to allow users to fetch coded blocks from multiple sources in order to finally decode the original content. In [10], the authors theoretically and practically demonstrate that coding can improve the throughput of network loads for content distribution in wireless networks. In [11], the authors investigate the storage allocation problem with network coding for proactive storage at storage-enable BSs in cellular network. In our work, we also consider to improve the caching diversity and content availability by applying network coding in the content placement phase.

Two classes of literature are highly related to our study. The first one is content caching systems in mobile networks. These works typically try to determine what to replicate on BS caches. In [11], the authors design algorithms to allocate storage capacity for one content in mobile network with cache-enabled BSs. They show that the complexity of the content placement problem is NP-Hard. In [12], the authors propose several caching policies for BS caching system in order to reduce the traffic throughput on the mobile gateway and backhaul links. In [13], the authors implement caches on small cells to mitigate traffic pressure on backhaul links. They study the optimum content placement problem to minimize the file downloading time. In [14], the authors investigate content replacement strategy on BS cache. The problem is modeled by a Markov decision process, and a distributed content placement algorithm is proposed. In [15], the authors propose a distributed content placement algorithm based on belief propagation to reduce the average content download delay in a mobile network with BS caches. However, these work do not consider BSs cooperations for caching contents and replying user requests.

The second class of literature is about the collaborative content placement problem which has been extensively studied in the traditional wired CDN systems. Typically, these works tackle the joint content placement and user request redirection within the range of CDN systems. Specifically, the joint content placement and user request redirection problem is known as NP-Hard. In [16], the authors consider the two problems separately at different time scales to reduce the problem complexity. In [4], the authors jointly address content placement and user redirection, subject to storage capacity and link bandwidth. However, the complexity of the problem prevents a fast efficient solution. The authors utilize the exponential potential function method to find sub-optimal solutions. In [5], the joint problem is studied in a hierarchical tree structure. The problem is solved optimally in a certain simplified scenario. In [17], cache replacement algorithm is designed through dynamic programming upon the assumption of a-priori knowledge of network parameters. In [18], the joint optimization problem is formulated to minimize content access delay in a general CDN architecture, and two heuristic algorithms are proposed.

These studies devise the joint content placement and request redirection schemes based on statistical network parameters, such as content popularity and user demand. However, these parameters are impractical to evaluate and collect on time. Moreover, the distribution of these parameters can vary, especially in the highly dynamic mobile network environment. For instance, user mobility leads to frequent change of user demands on BSs. To overcome these limitations, we devote ourselves to design online algorithms without requiring any a-priori knowledge on network parameters, which are more suitable for practical implementation in the dynamic mobile network environment.

There are works designing online algorithm for mobile network collaborative caching [3, 19]. In [19], the authors propose an online algorithm for the collaborative caching problem in multiple coordinated BSs without requiring knowledge about content popularity. However, their request scheduling machnism is solely based on content availability on BSs. In [3], the joint content placement and scheduling problem is studied in the context of wireless network. The authors design on-line algorithms by analyzing user request queues on each BS. However, the BSs workload and congestion level cannot be directly reflected by user requests queues. In fact, none of the above two works seeks to find joint content placement and user request redirection scheme for balancing traffic load and avoiding network congestion.

III. System Model

We consider a Mobile-CDN system, where storage is implemented on mobile network BS. There are N BSs in the system. We denote BS by n_i, where $i \in [1, \cdots, N]$. BS n_i is equipped with a storage of size S_i. The system operates in discrete time with time slot index $t \in \{0, 1, 2, \cdots\}$. Network parameters are time varying, however they are supposed to be unchanged during one time slot. Further, in the current time slot, the values of the parameters for future time slots are unknown. That is, we do not have a-priori knowledge on future network configurations and states.

The BSs cooperate in replying user content requests. Typically, we do not consider a centralized remote content server or content vault in the system. Thus, if the content required by a user is not stored in the local cache, the user requests will be redirected to other BSs. BSs can transmit data among them through the mobile network backhaul links. For example, in the Cloud-RAN (Cloud-Radio Access Network) architecture, BSs are regarded as interconnected through a central processor, thus inter-BS transmission is facilitated. We denote the transmission cost of one chunk from BS n_i to BS n_j at time t as $c_{ij}(t)$, for $i \in [1, \cdots, N]$ and $j \in [1, \cdots, N]$.

A. Content Placement and Request Redirection

The MNO aims to replicate a group of content on the BS storages. A content is a generic piece of information. For simplicity, we assume that all content have the same size (for example, a content can represent a video chunk in a VoD application). We denote each content by $k \in [1, \cdots, K]$. In order to ease the distributed storage of contents and to allow users to recover the content from multiple BSs caches, network coding is applied in the content placement phase. Content is coded into blocks and cached distributedly. Then, it is sufficient for the users to receive a certain number of blocks to decode the original content. We define the content placement decision variable $x_i^k(t)$, which represents the ratio of cached coded blocks over the number of blocks to decode the original content on BS n_i for content k at time slot t. Obviously, $0 \leq x_i^k(t) \leq 1, \forall i, k, t$. The storage capacity constraint requires each BS storage n_i to satisfy:

$$\sum_k x_i^k(t) \leq S_i, \quad \forall i, t. \tag{C1}$$

In BS cooperation, when the locally stored blocks are not sufficient to decode the original content, the system should decide where the user request should be redirected. A request redirection decision variable is defined as $y_{ij}^k(t)$, which represents the percentage of blocks that should be transmitted from BS n_i to BS n_j at time t for content k. Thus, $0 \leq y_{ij}^k(t) \leq 1, \forall i, j, k, t$. User redirection should comply with the content availability on BSs. Moreover, users should receive sufficient blocks to decode the original content. Thus, the following two constraints are imposed:

$$y_{ij}^k(t) \leq x_i^k(t), \quad \forall i, j, k, t, \tag{C2}$$

$$\sum_j y_{ji}^k(t) + x_i^k(t) \geq 1, \quad \forall i, k, t. \tag{C3}$$

Constraint (C2) restricts on the amount of content that could be supplied by each BS. Constraint (C3) means each BS should receive enough content to decode the original content. Please note that to make the problem feasible, a lower bound on storage capacity is that the aggregate storage capacity should be sufficient to at least store one copy of the coded blocks of all content.

B. Transmission Queue Stability

Each BS maintains a transmission queue, containing all data that should be sent to other BSs. Suppose the BS n_i's uplink capacity on backhaul link at time t is denoted as $B_i(t)$. Thus, $B_i(t)$ shows the transmission capacity of BS n_i to other BSs, and it represents the serving rate of the transmission queue on BS n_i. Then, the arriving rate for queue on BS n_i is the aggregate of user requests redirected to the BS: $\sum_k \sum_j y_{ij}^k(t)$. The states of the queues show the traffic load on the BSs, thus should be considered for the decision of request redirection.

The queue backlog associated to BS n_i at time t is denoted as $Q_i(t)$, which evolves as follows:

$$Q_i(t+1) = \max[(Q_i(t) - B_i(t)), 0] + \sum_k \sum_j y_{ij}^k(t). \tag{1}$$

The queue $Q_i(t)$ is defined as *strongly stable* if the following condition holds [20]:

$$\lim_{T \to \infty} \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}\{|Q_i(t)|\} < \infty.$$

The strong stability enforces bounded queue backlogs, thus this property avoids network congestions. Further, a multi-queue network is strongly stable if all the individual queues are strongly stable, which means that the N transmission queues are all bounded. For a queue to be stable, the average arriving rate should be no higher than the average serving rate. In the current paper, we suppose the overall system serving capacity is overprovisioned to transmit all user requests, thus stability is affected by user request redirection schemes.

IV. Problem Formulation and Transformation

A. The Stochastic Optimization Problem

The MNO is concerned with the transmission cost incurred by user request redirection. The transmission cost at time slot t is calculated as:

$$c(t) = \sum_k \sum_i \sum_j d_i^k(t) c_{ji}(t) y_{ji}^k(t)$$

The number of user requests issued on BS n_i for content k at time slot t is denoted as $d_i^k(t)$. Then, the problem is formulated as a stochastic optimization problem that minimizes the long

The long-term stochastic problem formulation

$$\min_{\{x_i^k(t)\},\{y_{ij}^k(t)\}} \overline{C} = \lim_{T\to\infty} \frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}\{c(t)\} \quad \text{(P1)}$$

$$\text{s.t.} \quad \sum_k x_i^k(t) \le S_i, \quad \forall i,k,t \quad \text{(C1)}$$

$$y_{ij}^k(t) \le x_i^k(t), \quad \forall i,j,k,t \quad \text{(C2)}$$

$$\sum_j y_{ji}^k(t) + x_i^k(t) \ge 1, \quad \forall i,k,t \quad \text{(C3)}$$

$$0 \le x_i^k(t) \le 1, \quad \forall i,k,t \quad \text{(C4)}$$

$$0 \le y_{ij}^k(t), \quad \forall i,j,k,t \quad \text{(C5)}$$

$$y_{ii}^k(t) = 0, \quad \forall i,k,t \quad \text{(C6)}$$

$$Q_i(t) \text{ is strongly stable}, \quad \forall i. \quad \text{(C7)}$$

term time average transmission cost, subjects to storage capacity, content integrity and network stability constraints (shown below).

B. Problem Transformation

We utilize the *Lyapunov drift-plus-penalty* method to solve the above stochastic optimization problem (P1) [20]. This method transforms the original problem into a series of static optimization problems, which minimize the drift-plus-penalty term in each time slot. In the following, we detail the process of this transformation, and discuss the rationale behind this method.

Let $\mathbf{Q}(t) = \{Q_i(t)\}$ be the vector of queue backlogs for time t. Then, we define the quadratic Lyapunov function for t as:

$$L(\mathbf{Q}(t)) \triangleq \frac{1}{2}\sum_i Q_i^2(t).$$

The quadratic Lyapunov function has the property of balancing traffic loads among BSs. Since it is calculated as the sum of the square of all queue backlogs, when it is large, at least one BS endures heavy traffic load. As a result, pushing Lyapunov function into a lower value ensures network stability, while at the same time achieves load balancing.

Without loss of generality, all queues are assumed to be empty when $t = 0$ such that $L(\mathbf{Q}(0)) = 0$. Define the one slot conditional Lyapunov drift $\Delta(\mathbf{Q}(t))$ as:

$$\Delta(\mathbf{Q}(t)) \triangleq \mathbb{E}\{L(\mathbf{Q}(t+1)) - L(\mathbf{Q}(t))|\mathbf{Q}(t)\}.$$

The Lyapunov drift-plus-penalty method seeks to minimize the drift-plus-penalty term $\Delta(\mathbf{Q}(t)) + V\mathbb{E}\{c(t)|\mathbf{Q}(t)\}$ at each time slot, such that the queue backlogs are continuously pushed towards a lower congestion state, whereas at the same time approaching to the optimization goal. We first show the upper bound of the drift term as:

$$\begin{aligned}\Delta(\mathbf{Q}(t)) &= \sum_i \mathbb{E}\{L(\mathbf{Q}(t+1)) - L(\mathbf{Q}(t))|\mathbf{Q}(t)\} \\ &\le B - \sum_i Q_i(t)B_i(t) \\ &\quad + \mathbb{E}\{\sum_i Q_i(t)\sum_k\sum_j y_{ij}^k(t)|\mathbf{Q}(t)\}.\end{aligned}$$

where B is a positive constant such that

$$B \ge \frac{1}{2}\mathbb{E}\{\sum_k\sum_i\sum_j {y_{ij}^k}^2(t)|\mathbf{Q}(t)\} + \frac{1}{2}\sum_i B_i^2(t)$$

Thus, the upper bound on the drift-plus-penalty term is derived as:

$$\begin{aligned}\Delta(\mathbf{Q}(t)) + V\mathbb{E}\{c(t)|\mathbf{Q}(t)\} &\le B - \sum_i Q_i(t)B_i(t) \\ &\quad + \mathbb{E}\{\sum_i Q_i(t)\sum_k\sum_j y_{ij}^k|\mathbf{Q}(t)\} \\ &\quad + V\mathbb{E}\{\sum_k\sum_i\sum_j d_i^k(t)c_{ji}(t)y_{ji}^k(t)|\mathbf{Q}(t)\}.\end{aligned}$$

Here, V is a tuneable positive parameter. Minimizing the drift-plus-penalty term in each time slot can ensure the network stability while at the same time optimize the long term optimization problem. By employing the concept of opportunistically minimizing an expectation, the right hand side of the drift-plus-penalty term is minimized by greedily minimizing:

$$\sum_k\sum_i\sum_j \left(Q_i(t) + Vd_j^k(t)c_{ij}(t)\right) y_{ij}^k(t)$$

Thus, the original problem (P1) can be solved equivalently by solving the following optimization problem (P2) at each time slot (we omit the term t for notation simplification):

$$\min_{x_i^k, y_{ij}^k} \quad \sum_k\sum_i\sum_j (Q_i + Vd_j^k c_{ij})y_{ij}^k \quad \text{(P2)}$$

$$\text{s.t.} \quad \text{(C1), (C2), (C3), (C4), (C5) and (C6).}$$

From this problem transformation, we can observe that user redirection decisions are made based on the backlog information of transmission queues. To solve Problem (P2), large queue backlog leads to less redirected user requests to ensure network stability. Moreover, the problem only depends on the current network parameters and queue state information (QSI), which permits the design of online algorithms.

Problem (P2) is formulated as a Linear Program (LP) which can be solved by a standard optimizer such as IBM ILOG CPLEX. However, the complexity of the problem prevents a fast computation of optimum solution. In the next section, we present our algorithm which efficiently solves the content placement and user request redirection problems.

V. Online Algorithm

In this section, we first present an efficient solution of Problem (P2). Then, we introduce our semi-distributed online algorithm, which jointly decides content placement and request redirection based on QSI updates. Then, we discuss its practical implementation relevance.

Problem (P2) is hard to solve. However, to timely reflect network traffic condition, time slot duration is required to be relatively short in our work. Thus, we aim to find an efficient way to solve the Problem (P2). In order to reduce the complexity of problem (P2), we first relax constraint (C3) by using Lagrangian relaxation. The corresponding Lagrangian function is given by:

$$L(\{x_i^k\},\{y_{ij}^k\},\boldsymbol{\lambda}) = \sum_k \sum_i \sum_j (w_{ij}^k - \lambda_j^k) y_{ij}^k - \sum_k \sum_i \lambda_i^k x_i^k + \sum_k \sum_i \lambda_i^k$$

where $w_{ij}^k = Q_i + V d_j^k c_{ij}$, and $\boldsymbol{\lambda} = \{\lambda_i^k\}$ is the Lagrange multipliers matrix for constraint (C3). Thus, the Lagrangian dual function is defined as:

$$g(\boldsymbol{\lambda}) = \min_{\{x_i^k\},\{y_{ij}^k\}} L(\{x_i^k\},\{y_{ij}^k\},\boldsymbol{\lambda}).$$

And the dual problem is optimizing the dual function subject to the dual variables:

$$\max_{\boldsymbol{\lambda}} \quad g(\boldsymbol{\lambda}) \quad \text{s.t.} \quad \boldsymbol{\lambda} \succeq 0. \tag{P3}$$

We further decompose Problem (P3) and separate it into a content placement subproblem and a user request redirection subproblem. These two problems can be efficiently solved.

A. Content Placement Subproblem

The content placement subproblem involves the content placement variable $\{x_i^k\}$. It is formulated as:

$$\max_{0\le \boldsymbol{x}\le 1} \quad \sum_k \sum_i \lambda_i^k x_i^k \quad \text{s.t.} \quad \sum_k x_i^k \le S_i \quad \forall i,k. \tag{P4}$$

This problem formulation is a typical Linear Knapsack Problem (LKP), which can be optimally solved by each BS in polynomial time by knowing the value of multipliers $\{\lambda_i^k\}$. By solving this problem, each BS obtains the optimum content placement $x_i^{k^*}$ for a given $\boldsymbol{\lambda}$.

B. User Request Redirection Subproblem

Then, the user request redirection subproblem is expressed as follows:

$$\min_{\substack{0\le y_{ij}^k \le x_i^{k^*} \\ y_{ii}^k = 0}} \quad \sum_k \sum_i \sum_j (w_{ij}^k - \lambda_j^k) y_{ij}^k. \tag{P5}$$

Algorithm 1 Joint Content Placement and Request Redirection Algorithm for each time slot

1: **Initialization**:
2: At each time slot, each BS sends its QSI to the CC.
3: Based on the network parameter observed in the current time slot, CC computes w_{ij}^k, $\forall i,j,k$.
4: The CC generates random initial $\lambda_i^k(s=0)$ and disseminates to BSs.
5: **while** not converge and $s < S_{MAX}$ **do**
6: **BS part**: for each BS n_i
7: Upon receiving $\lambda_i^k(s)$, solves (P4).
8: Sends $x_i^{k^*}$, $\forall k$ to CC.
9: **CC part**:
10: Upon receiving $\{x_i^{k^*}\}$, solves (P5) and obtains $\{y_{ij}^{k^*}\}$.
11: Based on $\{x_i^{k^*}\}$ and $\{y_{ij}^{k^*}\}$ updates $\{\lambda_i^k(s=s+1)\}$ according to Eq. (2).
12: Disseminates $\{\lambda_i^k(s=s+1)\}$ to BSs.
13: **end while**
14: According to the final $\{x_i^{k^*}\}$, each BS updates its stored content.
15: According to the final $\{y_{ij}^{k^*}\}$, CC coordinates user requests redirection.
16: All BSs update QSI according to Eq. (1).

By observing the Problem (P5), given the values of $\{\lambda_i^k\}$ and $\{x_i^{k^*}\}$, we find that the optimal solutions of $y_{ij}^{k^*}$ have the following structure:

- If $i=j$, $y_{ii}^{k^*}=0$.
- If $i\neq j$ and $w_{ij}^k - \lambda_j^k \ge 0$, $y_{ij}^{k^*}=0$.
- If $i\neq j$ and $w_{ij}^k - \lambda_j^k < 0$, $y_{ij}^{k^*}=x_i^{k^*}$.

In this way, we separately solve the original dual Problem (P3) by a sub-optimum solution for content placement and user request redirection. The dual variables $\{\lambda_i^k\}$ are updated based on the subgradient method:

$$\lambda_i^k(s+1) = [\lambda_i^k(s) + \alpha(1 - \sum_j y_{ji}^{k^*}(\lambda_i^k(s)) - x_i^{k^*}(\lambda_i^k(s)))]^+ \tag{2}$$

where s stands for the iteration number, α is the step size, $y_{ji}^{k^*}(\lambda_i^k(s))$ and $x_i^{k^*}(\lambda_i^k(s))$ are the solutions of the content placement decision variables and user request redirection variables for a given $\boldsymbol{\lambda}$, and $[\cdot]^+$ is the nonnegative orthogonal projection.

C. Semi-Distributed OJCPRR Algorithm

The above solution can be implemented through a semi-distributed online algorithm (as detailed in Algorithm 1). We name this algorithm as OJCPRR which stands for On-line Joint Content Placement and Request Redirection algorithm. For practical implementation, a central coordinator (CC) is required for performing computation and orchestrating information exchanges. The algorithm is executed periodically on both the CC part and BS part for each time slot based on the value of the current network parameters and BS QSIs.

In the initialization phase of each time slot, each BS first sends its current QSI and other parameter information to the CC. Based on these information, the CC computes the value of w_{ij}^k and generates random initial $\lambda_i^k(s=0)$, then it disseminates these information to all BSs.

Upon receiving $\lambda_i^k(s)$ from the CC, each BS solves the Problem (P4) to find ${x_i^k}^*$ for the current λ_i^k value, and sends it back to the CC. Based on $\{{x_i^k}^*\}$, the CC solves the Problem (P5) and obtains $\{{y_{ij}^k}^*\}$. Then, the CC updates $\lambda_i^k(s+1)$ and disseminates the new $\boldsymbol{\lambda}$ values to BSs. This process iterates until the computation converges or a maximum iteration number S_{MAX} is reached.

The final $\{{x_i^k}^*\}$ and $\{{y_{ij}^k}^*\}$ values determine the content placement and user request redirection for the current time slot. Finally, each BS updates its transmission queue state according to Eq. (1), and the algorithm goes into the next time slot.

VI. Evaluations

We build a simulation platform to evaluate the performance of our proposed online joint content placement and request redirection algorithm OJCPRR. We evaluate two sets of simulations: an illustrative small scale simulation and a real trace based large scale simulation.

A. Small Scale Simulation

In the small scale simulation, we emulated a network with 3 BSs running over $1,000$ time slots. The aim of this simulation is to investigate and illustrate the queue-aware feature of the algorithm OJCPRR in detail. The BS backhaul uplink capacity $\{B_i(t)\}$ is independent identically distributed over time slots. We assume that $\{B_i(t)\}$ follows Gaussian distribution with μ equals 550 MBps and σ^2 equals 50 for all BSs. These values are set to make sure that the aggregate serving rate of the system is sufficient to transmit all user requested content to avoid inherent unstable system.

The system aims to replicate 5 contents with each set to 200 MB. The storage size of each BS is set to 500 MB, thus one BS cannot store all content, and 2.5 contents should be fetched from other BSs by user requests redirection. According to this setting, the average transmission capacity of the system is sufficient for content transmission (7.5 content for each time slot). For each time slot, each BS serves 50 users. These users issue content requests according to a Zipf's content popularity distribution.

Further, we deliberately differentiate the transmission cost between BSs. The cost between BS1 and BS2 is set to 1, whereas the cost from BS3 to the other two BSs is set to 2. By doing so, we aim to investigate in detail the algorithm's performance on reacting to different transmission costs. Finally, we set the V parameter value used in the Lyapunov drift-plus-penalty term to 1.

The queue-aware nature is the main characteristics of our OJCPRR algorithm. Thus, we compare our algorithm to the traditional off-line cost-driven queue-unaware one, which decides content placement and request redirection solely based on the cost setting of the network [5]. Particularly, in each time slot, the traditional algorithm tries to solve the following problem:

$$\begin{aligned} \min_{x_i^k, y_{ij}^k} \quad & c(t) = \sum_k \sum_i \sum_j d_i^k(t) c_{ji}(t) y_{ji}^k(t) \\ \text{s.t.} \quad & \text{(C1) to (C6).} \end{aligned}$$

The optimum solution of this problem is obtained by the IBM ILOG CPLEX optimizer. Thus, this algorithm is queue-unaware that leads to instable system and arouse network congestion.

We first discuss the transmission cost of OJCPRR. Actually, the time average transmission cost obtained by our algorithm is 12.48, whereas the time average transmission cost obtained by the cost-driven queue-unaware algorithm is 10.66, which is lower than OJCPRR. That is because the queue-unaware algorithm tries to minimize transmission cost in each time slot. For OJCPRR, to avoid network congestion, user requests can be redirected to BS with higher transmission cost but shorter queue, which leads to higher transmission cost.

Then, we demonstrate the ability of the OJCPRR algorithm for stabilizing queues. In Fig. 1, we show the queue backlog variation of BS1 over all time slots obtained by the OJCPRR algorithm, whereas in Fig. 2, we show the result obtained by the queue-unaware algorithm of BS1. For OJCPRR, the backlog of the transmission queue for BS1 is strictly bounded (always below 800 MB), which implies strong stability for the transmission queue on BS1 and no network congestion occurred. However, for the queue-unaware algorithm BS1 experiences instable queue. The queue backlog steadily increases into over $130,000$ MB. This shows the advantages of our OJCPRR algorithm on avoiding congestion.

Then, we also compare the queue backlogs of the OJCPRR algorithm to that of the queue-unaware algorithm in Table. I. We first observe that for BS1 and BS2 OJCPRR obtains much lower average queue backlogs than the queue-unaware one, whereas for BS3 OJCPRR has higher queue backlog than that of the latter. The cost-driven queue-unaware algorithm avoids redirecting to BS3, regardless of the high congestion level of BS1 and BS2. For OJCPRR, more user requests are redirected to BS3, although this brings higher transmission cost, the queue backlogs are bounded for all the three BSs.

Finally, we look into details of the queue backlogs of the three BSs for the OJCPRR algorithm. In Fig. 3, we show the queue backlogs information for all the three BSs by the box-and-whisker plot, where the medium, the upper/lower quartile and the upper/lower whisker are shown. The average queue backlogs information for the three BSs is shown in Table. I and is not shown in the figure. Firstly, BS3 endures lower queue backlogs than the other two BSs. That is because BS3 is set to have higher transmission cost for sending content to the other two BSs. Consequently, for BS1 and BS2 it is better to redirect user requests to each other to lower down transmission cost, which brings higher backlog for these two BSs. Furthermore, BS1 and BS2 shares similar backlogs. This

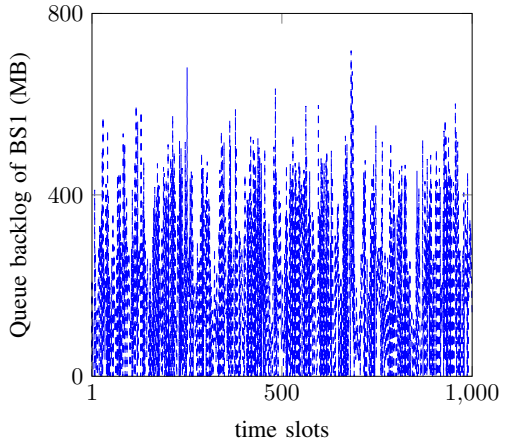

Fig. 1: Queue backlogs of BS1 of the queue-aware OJCPRR algorithm

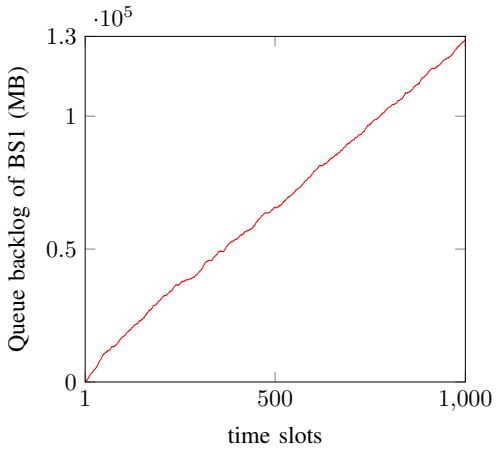

Fig. 2: Queue backlogs of BS1 for the queue-unaware algorithm

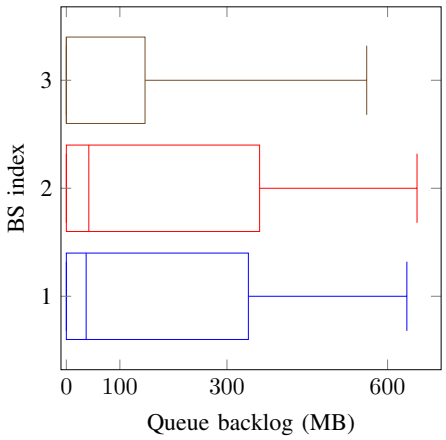

Fig. 3: Queue backlogs of BSs for the small scale simulation

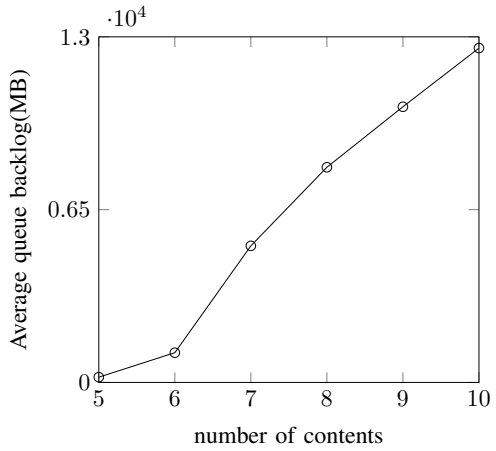

Fig. 4: Average queue backlogs for the large scale simulation

TABLE I: Average queue backlogs (MB)

	BS1	BS2	BS3
OJCPRR	164.9	169.5	93.4
queue-unaware	66087.6	71432.7	0.08

shows the advantages of our user request redirection algorithm on balancing traffic loads.

B. Large Scale Simulation

We also conduct a set of real-trace based large scale simulations. We utilize the real mobile network user activity traces on the date of 2014/07/16 in the city Rennes (France) as our simulation scenario. The top 60 populated BSs are chosen as the storage enabled BSs. The mobile users served by these BSs generate content request according to the Zipf's content popularity distribution. The transmission cost between BSs are uniformly distributed between 1 and 2. We vary the number of serving content from 5 to 10 to represent different level of workload. The other simulation settings (such as the BS backhaul uplink capacity $\{B_i(t)\}$, etc.) remains the same as in the small scale simulation.

For the large scale simulation, the BS average queue backlogs of different number of serving content is shown in Fig. 4. As the provisioning of the serving capacity of the system remain unchanged, when the number of serving content increases, the congestion level of the system also increases. When the system transmits 5 contents, the average queue backlogs is 197 MB, whereas for 10 contents, the average queue backlog is increased to $12,577$ MB. Actually, for the serving traffic load exceeding the transmission capacity of the system, the queues are instable even with the congestion avoidance OJCPRR algorithm.

VII. Conclusions

In this paper, we investigated the joint optimization problem for content placement and user request redirection in the BS-based mobile CDN system. We utilize the Lyapunov method to solve a long-term stochastic optimization problem and design on-line algorithms which could be practically implemented. Comparing to the traditional queue-unaware algorithm, our

solution avoids traffic congestion and balances work loads. The current work has some limitations. For future work, as shown in the large scale simulation, the average system serving rate should be large enough to be able to transmit all user requests. To prevent such inherent instable case, in the next steps, we would like to introduce the content server with large transmission capacity but higher transmission cost, and investigate the performance of the algorithm in this scenario. Moreover, the current algorithm solves a sub-optimum solution for the joint optimization problem in each time slot. In the future, we aim to design efficient algorithm with provable bounds.

VIII. Acknowledgement

This work was supported by the Central Universities of China Fundamental Research Funds from Xidian University under Grant 20101156084, the China Postdoctoral Science Foundation under Grant 2015M582613, the National Natural Science Foundation of China under Grant 61602362 and 61471287, and the 111 Project B08038.

References

[1] S. B. H. Said, M. R. Sama, K. Guillouard, L. Suciu, G. Simon, X. Lagrange, and J. Bonnin, "New control plane in 3gpp LTE/EPC architecture for on-demand connectivity service," in *Proc. of IEEE Conf. CloudNet*, 2013, pp. 205–209.

[2] A. Gharaibeh, A. Khreishah, B. Ji, and M. Ayyash, "A provably efficient online collaborative caching algorithm for multicell-coordinated systems," *CoRR*, vol. abs/1509.02911, 2015.

[3] N. Abedini and S. Shakkottai, "Content caching and scheduling in wireless networks with elastic and inelastic traffic," *IEEE/ACM Transactions on Networking*, vol. 22, no. 3, pp. 864–874, June 2014.

[4] D. Applegate, A. Archer, V. Gopalakrishnan, S. Lee, and K. K. Ramakrishnan, "Optimal content placement for a large-scale vod system," in *Proc. of the 6th International Conference Co-NEXT '10*, 2010, pp. 4:1–4:12.

[5] S. Borst, V. Gupta, and A. Walid, "Distributed caching algorithms for content distribution networks," in *Proc. of IEEE INFOCOM 2010*, March 2010, pp. 1–9.

[6] F. Yousaf, M. Liebsch, A. Maeder, and S. Schmid, "Mobile cdn enhancements for qoe-improved content delivery in mobile operator networks," *IEEE Network*, vol. 27, no. 2, pp. 14–21, March 2013.

[7] M. Liebsch and F. Yousaf, "Runtime relocation of cdn serving points - enabler for low costs mobile content delivery," in *Proc. of 2013 IEEE Wireless Communications and Networking Conference (WCNC)*, April 2013, pp. 1464–1469.

[8] M. Almashor, I. Khalil, Z. Tari, A. Zomaya, and S. Sahni, "Enhancing availability in content delivery networks for mobile platforms," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 8, pp. 2247–2257, Aug. 2015.

[9] J. Liu, Q. Yang, and G. Simon, "Optimal and practical algorithms for implementing wireless cdn based on base stations," in *Proc. of the IEEE VTC-Spring*, May 2016.

[10] M. A. Maddah-Ali and U. Niesen, "Coding for caching: fundamental limits and practical challenges," *IEEE Communications Magazine*, vol. 54, no. 8, pp. 23–29, August 2016.

[11] J. Gu, W. Wang, A. Huang, and H. Shan, "Proactive storage at caching-enable base stations in cellular networks," in *Proc. of 2013 IEEE 24th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, Sept 2013, pp. 1543–1547.

[12] H. Ahlehagh and S. Dey, "Video-aware scheduling and caching in the radio access network," *IEEE/ACM Transactions on Networking*, vol. 22, no. 5, pp. 1444–1462, Oct. 2014.

[13] K. Shanmugam, N. Golrezaei, A. Dimakis, A. Molisch, and G. Caire, "Femtocaching: Wireless content delivery through distributed caching helpers," *IEEE Transactions on Information Theory*, vol. 59, no. 12, pp. 8402–8413, Dec. 2013.

[14] J. Gu, W. Wang, A. Huang, H. Shan, and Z. Zhang, "Distributed cache replacement for caching-enable base stations in cellular networks," in *Proc. of 2014 IEEE International Conference on Communications (ICC)*, June 2014, pp. 2648–2653.

[15] J. Liu, B. Bai, J. Zhang, and K. B. Letaief, "Content caching at the wireless network edge: A distributed algorithm via belief propagation," in *Proc. of 2016 IEEE International Conference on Communications (ICC)*, May 2016, pp. 1–6.

[16] J. He, H. Zhang, B. Zhao, and S. Rangarajan, "A collaborative framework for in-network video caching in mobile networks," *CoRR*, vol. abs/1404.1108, 2014.

[17] J. Wu and B. Li, "Keep cache replacement simple in peer-assisted vod systems," in *Proc. of IEEE INFOCOM 2009*, April 2009, pp. 2591–2595.

[18] P. Amani, S. Bastani, and B. Landfeldt, "Towards optimal content replication and request routing in content delivery networks," in *Proc. of 2015 IEEE International Conference on Communications (ICC)*, June 2015, pp. 5733–5739.

[19] A. Gharaibeh, A. Khreishah, B. Ji, and M. Ayyash, "A provably efficient online collaborative caching algorithm for multicell-coordinated systems," *IEEE Transactions on Mobile Computing*, vol. 15, no. 8, pp. 1863–1876, 2016.

[20] M. J. Neely, *Stochastic Network Optimization with Application to Communication and Queueing Systems*. Morgan & Claypool, 2010.