2020 IEEE 40th International Conference on Distributed Computing Systems (ICDCS) | 978-1-7281-7002-2/20/$31.00 ©2020 IEEE | DOI: 10.1109/ICDCS47774.2020.00106

Privacy-Preserving Distributed Edge Caching for Mobile Data Offloading in 5G Networks

Yiming Zeng, Yaodong Huang, Ji Liu, Yuanyuan Yang
Department of Electrical and Computer Engineering
Stony Brook University, Stony Brook, NY 11794, USA
{yiming.zeng, yaodong.huang, ji.liu, yuanyuan.yang}@stonybook.edu

***Abstract*—Distributed edge caching has drawn great attention with the fast development of smart edge devices. Caching popular contents in the edge can reduce latency and improve the quality of service of edge mobile users. Meanwhile, the data privacy in the edge is critical to preserve the privacy of individual users and devices. How to jointly determine the caching and routing policy in the edge network in a distributed manner and simultaneously design the proper privacy preserving mechanism are challenging. We tackle these challenges in two progressive steps. First, we design a distributed algorithm which can achieve the global optimum. Second, we propose a privacy-preserving mechanism based on differential privacy and prove the privacy guarantee. We conduct extensive numerical simulations based on real-world requests to evaluate the performance of the proposed distributed algorithm and the privacy mechanism. Results highlight a significant improvement of the proposed distributed algorithm while only up to 10.1% of the total serving cost increased by the privacy mechanism.**

***Keywords*—Edge caching, Differential privacy, Distributed algorithm, 5G networks**

I. Introduction

The rapid development of edge devices such as smart phones and tablets brings an unprecedented network traffic for mobile services such as social media and live video streaming. The tremendous network traffic threatens to drain the capacity of the cellular network. To improve the quality of service of mobile users (MUs), the fifth generation cellular network technology (5G) is now being widely deployed all over the world. The expected transmission speed is roughly 20 times faster than what is possible with 4G; meanwhile, the latency can drop into single-digit milliseconds, making lag times nearly impossible to detect [1]. 5G network contains a large number of small base stations (SBSs) which are connected with the core network and placed very closed to MUs. Caching popular contents in the edge SBSs is a key technique in 5G to deliver the promise. Edge caching can greatly reduce the traffic of the back-hual links.

Distributed edge caching has been considered as a practical mechanism in the cellular network especially in the 5G network. SBSs are equipped with high-performance edge computing equipments for decision making to serve MUs independently. The base station (BS) of the core network does not need to deal with huge amount of requests from MUs directly, hence the huge computing overhead can not make the BS the system bottleneck. SBSs deployed in the edge are much closer to MUs compared with the BS, the transmission delay and the extra transmission power can be saved for the network. The real systems have been developed. For example, LinkEdge developed by Alibaba Cloud (2018) and Amazon Web Service (AWS) Greengrass (2017) can enable SBSs to conduct caching, routing and computing tasks in the edge [2].

Many distributed caching models have been studied in previous work. Belief propagation based algorithms are proposed in [3], [4]. In [5], [6], distributed methods based on consensus or convex optimization are applied with the performance guarantee. The learning methods are studied in [7], [8], but the performance can not be guaranteed. All the papers above focus on the specific case about caching problem neglecting how to determine the routing from SBSs to MUs with the bandwidth limitation. In this paper, we consider a more general and basic model considering both caching and routing of the network which are two fundamental functions for the content delivery in the 5G network.

Furthermore, the privacy of the edge network data is important yet seldom considered in previous studies. The privacy in the edge will be compromised by releasing personal and social data to some groups or companies such as e-commerce sites, rating services, search engine and location services without permission [9]. In the 5G network, there are three key sensitive types of data the requests of MUs, the caching policy and the routing policy when consider the content delivery issue. The information of requests are sent to the SBSs which can serve the requests and the BS. The information of requests can be accessed by all SBSs including the BS in the network, and could even be published by the BS. For instance, the number of views and comments can be accessed from videos websites. The information about requests especially the number of requests is a one time-shot and not so sensitive compared with the caching and routing policy. Caching policy is the commercial decision of the SBS which contains the commercial and technique information such as which content to be cached, cache size and computing ability. However, the caching policy is kept by the SBS without sharing in the network, and less possible to be accessed by attackers. Different from the caching policy kept by individual SBSs, routing policy is exchanged between the BS and individual SBS timely to fully serve MUs requests. Therefore, the information of routing policy could be accessed by attackers in the duration of the data transmission. Routing

2575-8411/20/$31.00 ©2020 IEEE
DOI 10.1109/ICDCS47774.2020.00106

policy does not only include the private information of MUs such as locations, the contents preference, it also contains the private information of SBSs including the channel, the memory size and the computing ability which are commercial private information. Especially when SBSs belong to different wireless service providers, the routing policy can not be accessed by other SBSs.

Recently, the privacy of data has been drawing great attention in other domains, especially in the data mining and the distributed learning [10]–[12]. Differential privacy [13], [14] can constitute the privacy guarantee for gathered data in the implementation of algorithms. It has been applied in Google to protect the privacy of users [15] from endless stream of concerning hacks. Several studies are conducted to protect the privacy of Internet of Things (IoT) in the edge [16]–[19]. The data privacy in the edge caching should also be carefully considered.

In this paper, we aim to answer the following questions:

(1) How to design the caching and routing policy of network in the distributed manner? And (2) how to ensure the privacy of the network?

The key challenges are:

- The optimization problem about jointly determining the caching and routing policy involves integers, which is a NP-hard problem. It is challenging in the centralized manner with all the information available. In the distributed manner, SBSs do not communicate with each other. Therefore, the computational distributed solution need to be carefully designed.
- The privacy of SBSs information is significant, while ensure information exchange in the network and do not violent individual MU and SBS in formation at the same time is difficult. Simply aggregate results of analysis may not provide sufficient protection. Hence, design a privacy protect mechanism is necessary and challenging.

By addressing these challenges, the paper has the following contributions:

- We propose a distributed algorithm which jointly determines both caching and routing policies of the network and is proved to converge to the global optimal point.
- We design a mechanism using differential privacy [13] for the privacy of the routing information from different SBSs, and analyze the performance of the proposed distributed algorithm with the performance guarantee. To the best of our knowledge, this is the first paper which considers the privacy problem in edge caching problems, and implements the differential privacy mechanism to preserve the privacy of individual MUs and SBSs in an edge network.
- We evaluate the performance of the proposed distributed algorithm and the privacy preserving mechanism based on the real-world trace. Results highlight that the proposed distributed algorithm can significantly reduce the system cost. The privacy mechanism can reduce the system cost by 17.3% compared with the classical algorithm, and only 6.6% on average more than the optimum.

TABLE I
TABLE OF NOTATIONS USED IN THE PROBLEM FORMULATION

Notation	Definition
$\mathcal{N}$	Set of SBS $\mathcal{N} = \{1, 2, \ldots, N\}$
$\mathcal{U}$	Set of MU groups $\mathcal{U} = \{1, 2, \ldots, U\}$
$\mathcal{F}$	Set of files $\mathcal{F} = \{1, 2, \ldots, F\}$
Λ	MUs requests matrix
λ_{uf}	Demand of MU u for content f
$\mathcal{L}$	Connectivity matrix among the MUs and SBSs
l_{nu}	The connectivity between SBS n and MU u, $l_{nu} \in \{0, 1\}$
X	Set of caching variables $X = (x_{n,f})_{n\in\mathcal{N}, f\in\mathcal{F}}$
Y	Set of load balancing variables of SBS $(y_{nuf})_{n\in\mathcal{N}, u\in\mathcal{U}, f\in\mathcal{F}}$
C_n	Storage capacity of SBS n
B_n	Bandwidth capacity of SBS n
$\hat{d}_u$	Weighted transmission parameter to BS of the classes MUs u
d_{nu}	Weighted transmission parameter to SBS n of the classes MUs u
δ	The Laplace component factor
ϵ	The privacy budget
μ_n	The set of Lagrange multiplier

The rest of paper is organized as follows. Section II describes the system model and introduces the problem formulation. In Section III, we propose a distributed algorithm and proof its convergence to the optimum. The privacy mechanism is illustrated in Section IV, and we analysis the performance of the proposed privacy mechanism. Section V provides the numerical results. We discuss related work in Section VI, followed by conclusion in Section VII.

II. SYSTEM MODEL AND PROBLEM FORMULATION

In this section, we introduce the network model and formulate the problem as a mix-integer optimization, to minimize the total serving cost of system by determining the caching and routing policy jointly. Important notations are summarized in Table I.

A. System Model

The system model is depicted in Fig. 1. We study the downlink operation of the 5G network with one BS. Note that our analysis can be easily extended for multiple BSs. The BS can serve MUs in the total area and coordinate SBSs. Let the set $\mathcal{N} = \{1, 2, \cdots, N\}$ denote SBSs. The coverage of SBSs are overlapping in real cases, therefore, MUs in the covered area can be served by one SBS or multiple SBSs. The MU set $\mathbf{u}$ indicates the number of MUs at the same location seeing that the large geographical scale of network. We view the MUs in the same location as one group MU denoted as u. The set of MU groups is denoted as $\mathcal{U} = \{1, 2, \cdots, U\}$. $l_{nu} \in \{0, 1\}$ denotes the connectivity between SBS n and MU u, if $l_{nu} = 1$, MU u can be served by the SBS n, otherwise $l_{nu} = 0$. $\mathcal{L}$ denotes the connection metric of all l_{nu}.

The BS offers a set of $F = \{1, 2, \cdots, F\}$ of contents. The size of contents is assumed to be the same as 1 which

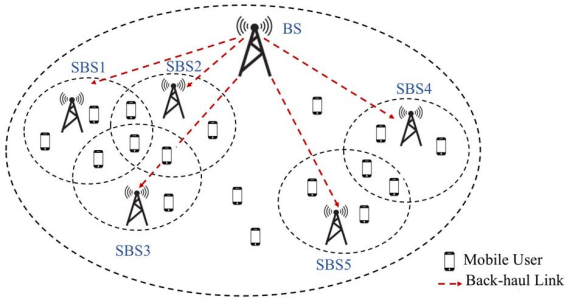

Fig. 1. An example of our proposed system model.

is reasonable for the reason that the content can be divided into blocks with the same size in the real system [20].

The mean requests arrival rate is denoted as λ_{uf}, the request from MU u for the content f. λ_{uf} could be greater than one considering that the content f can be requested multiple times by users in the MU group u. Denote by Λ the matrix of all λ_{uf}. To minimize the total serving cost of the network, it is preferable for SBSs to serve MUs directly seeing that SBSs are placed very close to MUs, the transmission delay is very small compared with the transmission delay from the BS. Furthermore, the energy cost can also be reduced a lot due to the less transmission power needed for SBSs and congestion in the back-hual link.

However, the SBS has limited memory resources for caching and bandwidth resources for contents transmission, it is lessly possible for SBSs to serve all MUs requests in the edge especially in peak period. Therefore, for each SBS, how to utilize the limited caching and routing resources needs to be carefully designed to minimize the total serving cost of the network. Specifically, SBSs seek values for two key parameters, caching policy and routing policy as follows,

- Caching policy: $x_{nf} \in \{0,1\}, \forall n \in \mathcal{N}, f \in \mathcal{F}$. x_{nf} is an integer variable indicates that whether the content f is cached in the SBS n or not. Caching policy is limited by memory resources of SBSs, i.e.,

$$\sum_{f \in \mathcal{F}} x_{nf} \leq C_n, \forall n \in \mathcal{N}. \tag{1}$$

  where C_n is the cache size of the SBS n.
- Routing policy: $y_{nuf} \in [0,1], \forall n \in \mathcal{N}, u \in \mathcal{U}, f \in \mathcal{F}$. y_{nuf} represents the portion of the requested content f served by SBS n. Routing policy of the SBS is limited by the caching policy and the bandwidth size. Formally,

$$y_{nuf} \leq x_{nf}, \forall n \in \mathcal{N}, u \in \mathcal{U}, f \in \mathcal{F} \tag{2}$$

$$\sum_{f \in \mathcal{F}} \sum_{u \in \mathcal{U}} y_{nuf} \cdot \lambda_{uf} \leq B_n, \forall n \in \mathcal{N}. \tag{3}$$

(2) indicates that the content needs to be cached ahead to serve MUs requests. In (3), B_n denotes the bandwidth size of SBS n. (3) indicates that total traffic cannot exceed the bandwidth of the SBS. Additionally, for the MU u, the total portion of content can not be served by SBSs repeatedly, formally,

$$\sum_{n \in \mathcal{N}} y_{nuf} \cdot l_{nu} \leq 1. \tag{4}$$

B. Problem Formulation

Our objective is to minimize the total serving cost of the network. The total serving cost is denoted as $f(\cdot)$ which can be decomposed as two parts.

The first part is the serving cost for SBSs to serve MUs requests directly denoted as $f_1(\cdot)$. The serving cost is generated by the energy consumption for the contents transmission, e.g., the spectrum and the transmission power, etc. It is the summation cost of N individual SBSs. For the SBS n, the serving cost is denoted as $f_{1n}(\cdot)$ which is counted on the portion of requests served for different MUs. Besides the number of requests, another important factor is the location of MUs. For instance, MUs located far away from the SBS need larger power to transmit requested contents. To clarify this, we denote $d_{nu} \geq 0$ as the weighted parameter.

In this paper, we assume $f_{1n}(\cdot)$ is non-decreasing about y_{nuf} and convex in all $y'_{nuf}s$. This has been widely applied in previous work. In [21], the author models the energy consumption cost function as a linear function of the total transmission power of base stations. Formally, we give a representative formulation of the cost function $f_{1n}(\cdot)$,

$$f_{1n}(y) = \sum_{u \in \mathcal{U}} \sum_{f \in \mathcal{F}} d_{nu} y_{nuf} \cdot l_{nu} \lambda_{uf}. \tag{5}$$

Hence, the total serving cost from SBS is as follows,

$$f_1(y) = \sum_{n \in \mathcal{N}} f_{1n}(y).$$

The second part is the serving cost for the BS to serve requests from MUs directly denoted as $f_2(\cdot)$. The distance parameter from the BS to MU u is denoted as $\hat{d}_u$. $\hat{d}_u$ is greatly larger than d_{nu} for any SBS $n \in \mathcal{N}$ seeing that the BS is located further away to MUs than SBSs placed in the edge. This makes the serving cost for the BS very large. Hence, to minimize the network total serving cost, the strategy is to let SBSs serve MUs as much as possible, the BS only needs to serve the remaining portion that SBSs are not able to due to the cache and bandwidth limitation. Similar with $f_1(\cdot)$, $f_2(\cdot)$ is continuously convex about y_{nuf}, the difference between $f_1(\cdot)$ and $f_2(\cdot)$ is that $f_2(\cdot)$ is non-increasing about y_{nuf}.

The cost function can be alternative with the properties mentioned above, we apply a representative one:

$$f_2(y) = \sum_{u \in \mathcal{U}} \hat{d}_u \sum_{f \in \mathcal{F}} \left(1 - \sum_{n \in \mathcal{N}} y_{nuf} \cdot l_{nu}\right) \lambda_{uf}. \tag{6}$$

Formally, the problem formulation can be derived as follows,

$$\min_{x,y} f(y) = f_1(y) + f_2(y), \tag{7}$$

$$s.t.(1),(2),(3),(4),$$

$$x_{nf} \in \{0,1\},\ \forall n \in \mathcal{N}, f \in \mathcal{F}, \tag{8}$$

$$y_{nuf} \in [0,1],\ \forall n \in \mathcal{N}, u \in \mathcal{U}, f \in \mathcal{F}. \tag{9}$$

The key challenges to solve this problem are as follows,

- Caching variable x_{nf} is an integer variable, which makes the problem as a NP-hard problem [22] and can not be solved directly.
- The constraint (4) couples all the routing policy of SBSs together, how to solve the problem distributedly is tricky.

III. Distributed Algorithm Design

In this section, we propose a distributed algorithm and prove its convergence to the global optimum. The similar problem has been successfully addressed in a centralized way with all the information of the network available in [22]. This fits for the network that all SBSs are owned to the BS with the real-time synchronization. However, a strong case in the real wireless communication network is that SBSs belong to various wireless communication companies. The BS is the content provider as the core Internet which serves contents to SBSs, and also serves MUs. The information of different companies cannot be shared with each other. For example, Netflix, a famous video provider, not only owns the core cache devices to serve all MUs, but also cooperates with wireless communication companies by caching videos on their SBSs to reduce the latency and improve the service of quality. Netflix coordinates SBSs from various companies by Cache Control Services (CCS) to serve MUs. If the request can not be satisfied, the core network of Netflix will send the requested video directly [23]. These SBSs belong to independent companies and the private commercial information e.g., the routing policy, can not be shared with each other. For simplicity, we assume that the MU can be served at most by one SBS from one individual company. According to the real case, the wireless communication company does not place multiple SBSs at the same location because of the signal interference and the equipment cost. To meet this need, the algorithm presented below is distributed and allows individual SBSs optimize their decisions based on partial information separately.

In the distributed network, SBSs do not communicate with others. However, the BS needs to aggregate routing policies from SBSs to serve remaining parts of requested contents from MUs, i.e. $1 - \sum_{n\in\mathcal{N}} y_{nuf}$.

Motivated by the observation that the problem can be separated in n with y_{nuf} and x_{nf}, except the constraint (4) and the constant component $\sum_{u\in\mathcal{U}}\sum_{f\in\mathcal{F}} \hat{d}_u \cdot \lambda_{nf}$. Furthermore, the BS can aggregates routing policies from SBSs. This allows the SBS n iteratively to determine the optimal caching policy x_{nf} and routing policy y_{nuf}, and communicate the intermediate routing policy with the help of the BS.

To highlight the separation among SBSs, we formulate the sub-problem P_n for the SBS n as:

$$\begin{aligned} P_n : \min_{y_n} f_n(y_n) = &\sum_{u\in\mathcal{U}}\sum_{f\in\mathcal{F}} d_{nu} y_{nuf} l_{nu} \lambda_{uf} + \\ &\sum_{u\in\mathcal{U}} \hat{d}_u \sum_{f\in\mathcal{F}} \left(1 - (y_{nuf} + y_{-n}) \cdot l_{nu}\right) \lambda_{uf} \end{aligned} \tag{10}$$

$$s.t., (8), (9),$$

$$s.t., \sum_{f\in\mathcal{F}} x_{nf} \le C_n, \tag{11}$$

$$y_{nuf} \le x_{nf}, \forall u \in \mathcal{U}, f \in \mathcal{F}, \tag{12}$$

$$\sum_{f\in\mathcal{F}}\sum_{u\in\mathcal{U}} y_{nuf} \cdot \lambda_{uf} \le B_n. \tag{13}$$

where

$$y_{-n} = \sum_{i<n} y_{iuf} + \sum_{i>n} y_{iuf}. \tag{14}$$

y_{-n} denotes the aggregation routing policy except the SBS n.

As we mentioned above, the BS can aggregate the routing policy from all SBSs to serve the remaining portion of requested contents. For the individual sub-problem P_n, SBS n does not need to access the specific routing policy of each SBS, the BS only needs send SBS n the aggregated routing information y_{-n}, and SBS n can not recover any information of individual SBS directly.

We address how to solve the sub-problem P_n first. Before we solve the P_n, we assume that y_{-n} as the input which is known before. The detailed update of y_{-n} is described in the distributed algorithm later.

We need to tackle two challenges to solve P_n. First, the caching variable x_{nf} is an integer, which makes the optimization not a continuously convex problem. It can not be solved directly. Second, caching variable x_{nf} and routing variable y_{nuf} is coupled together in the constraint (2).

To address two challenges, we relax the constraint (2) by introducing the set of Lagrange multipliers μ_n to decouple the constraint (2):

$$\mu_n = (\mu_{uf} \ge 0 : \forall u \in \mathcal{U}, f \in \mathcal{F}), \tag{15}$$

.

When relaxing the the constraint (2) by introducing μ_n, the new dual Lagrange function is:

$$L(y_n, \mu_n) = f_n(y_n) + \sum_{u\in\mathcal{U}}\sum_{f\in\mathcal{F}} \mu_{uf}(y_{nuf} - x_{nf}) \tag{16}$$

Then, the Lagrange relaxation problem is:

$$\max_{\mu_n} \min_{y_n} L(y_n) \tag{17}$$

$$s.t.(8),(9),(11),(14),(15).$$

This can be solved in an iterative manner, using a dual Lagrange decomposition method [24], [25]. The problem is further decomposed two sub-problems as the caching sub-problem and the routing sub-problem.

In the caching sub-problem, it only contains caching variables. The caching variable x_{nf} is relaxed from $\{0,1\}$ to be continues i.g., [0,1]. Hence, the optimization is strictly continuous and convex. We also prove later that after the relaxation, we can still derive the integral and optimal solutions. Formally, the caching sub-problem is defined as follows,

$$\max_{x_n} \sum_{f\in\mathcal{F}} \sum_{u\in\mathcal{U}} x_{nf}\mu_{nuf} \tag{18}$$

$$s.t.(11),(15)$$

$$x_{nf} \in [0,1], \forall f \in \mathcal{F}. \tag{19}$$

As we mentioned above, the caching sub-problem is a continuously convex problem which can be solved by the standard convex method [25].

Similar with the caching sub-problem, the routing sub-problem only contains the routing variable y_{nuf}. The routing sub-problem is defined as follows,

$$\min_{y_n} f_n(y_n) + \sum_{u\in\mathcal{U}} \sum_{f\in\mathcal{F}} \mu_{uf} y_{nuf}, \tag{20}$$

$$s.t.(9),(14),(15).$$

y_{nuf} is the continuous variable, and the routing sub-problem is convex about y_{nuf}, the standard convex solution can be applied [25].

For the dual variable, due to the discrete caching variable, the dual function can not be differential. Therefore, we apply a sub-gradient method [25] to ensure the convergence to the optimal solution of the relaxed problem. The iteration of the algorithm is denoted as k. In each iteration of k, the dual variable is updated as follow,

$$\mu_{uf}(k+1) = [\mu_{uf}(k) + \eta(k) g_{nf}(k)]^{+}, \tag{21}$$

where $\eta(k)$ denotes the step size, $g_{nf}(k)$ is the subgradient, and $[]^{+}$ indicates the projection. To ensure the convergence of the algorithm, from [[25], Chap.8], $\lim_{t\to\infty} \eta(t) = 0$ and $g_{nf}(k) \leq \infty$. There are multiple choices, and we select the following ones,

$$\eta(k) = \frac{1}{1+\alpha k} \tag{22}$$

and $g_{nf}(k)$ is chosen as the value of the relaxed inequality of the constraint, formally,

$$g_{nf}(k) = y_{nuf}(k) - x_{nf}(k) \tag{23}$$

where $y_{nuf}(k)$ and $x_{nf}(k)$ is the solution of caching and routing subproblem of iteration k.

In each iteration k, $y_{nuf}(k)$ and $x_{nf}(k)$ are derived by solving caching and routing sub-problems, the dual variable is updated by (21). The convergence can be guaranteed by (22) and (23) according to [25].

Note that when solving the caching subproblem, the integral caching variable x_{nf} is relaxed from $\{0,1\}$ to $[0,1]$, the derived optimal solution may not be decimal which is not feasible to the original problem. In this case, we prove that even after the relaxation, we can still acquire the optimal and integral caching policy.

Theorem 1. *The optimal solution of caching subproblem after the relaxation is integral.*

Due to the space limitation, we do not elaborate the proof of Theorem 1 here, and will provide the proof in an expanded version of this paper. We implement properties of totally unimodular matrix in the linear programming which can guarantee the integral and optimum after the relaxation.

From Theorem 1, the caching problem can obtain optimal results when relaxing the integer constraint. Then, the optimum of the sub-gradient presented above can be guaranteed [25].

After solving the sub-problem for the SBS n, we now introduce the distributed algorithm. The iteration of the algorithm is indexed by τ. Each iteration is divided into N phases. To start with, in the iteration $\tau = 0$, all SBSs determine their caching and routing policy independently based on MUs requests by solving the individual optimization P_n assuming that $y_{-n}(\tau = 0) = 0$. In the phase n of the iteration τ, SBS n determines the caching and routing policy of itself, by minimizing over its own variable x_{nf} and y_{nuf} :

$$\begin{aligned} P_n : \min_{y_n} f_n(y_n)(\tau) = & \sum_{u\in\mathcal{U}} \sum_{f\in\mathcal{F}} d_{nu} y_{nuf}(\tau) l_{nu} \lambda_{uf} + \\ & \sum_{u\in\mathcal{U}} \hat{d}_u \sum_{f\in\mathcal{F}} \left(1 - (y_{nuf}(\tau) + y_{-n}(\tau)) \cdot l_{nu}\right) \lambda_{uf} \end{aligned} \tag{24}$$

where $y_{-n}(\tau)$ is updated as follow:

$$y_{-n}(\tau) = \sum_{i<n} y_{iuf}(\tau) + \sum_{i>n} y_{iuf}(\tau - 1). \tag{25}$$

To compute $y_{-n}(\tau)$, SBS n needs not to access exact routing policies from other SBSs. Instead, the BS gathers the routing policy from SBSs, and calculates the aggregation routing policy for SBSs in every phase n of the iteration τ. Then the BS sends the aggregation routing policy to the SBS. Each SBS processes the step one by one in each iteration of different phases until the convergence of the algorithm. The distributed algorithm is summarized in the Algorithm 1.

Algorithm 1 Distributed Updating Algorithm

Require: initial feasible solution x_0 and y_0, T, Λ, d_{nu}, d_u, accuracy level γ, the maximum iteration times T.
Ensure: The optimal solution x^*, y^* and the corresponding optimal cost
1: **while** $\frac{f(y)(\tau)-f(y)(\tau-1)}{f(y(\tau))} > \gamma$ and $\tau \leq T$ **do**
2: **for** $n \in \mathcal{N}$ **do**
3: Solve the subproblem P_n to derive $x_{nf}(\tau)$ and $y_{nuf}(\tau)$,
4: Upload the routing policy $y_{nuf}(\tau)$ to BS, update the y ,
5: Broadcast the aggregated load $\sum_{n\in\mathcal{N}} y_{nuf}$ to all SBSs.
6: **end for**

Next, we prove the convergence and the optimality of the Algorithm 1. Algorithm 1 is motivated by the Gauss-Sedel iteration. The convergence and the optimality of the Algorithm is guaranteed by the Theorem 2.

Theorem 2. *Let $(x(\tau), y(\tau))$ be the sequence generated by the distributed algorithm. Then, starting from any initial feasible pair (x_0, y_0),*

(i) *$f(x(\tau), y(\tau))$ converges to the optimal value.*
(ii) *Any limiting point of the sequence $(x_n(\tau), y_n(\tau))$ is an optimal solution, for all SBS $n \in \mathcal{N}$.*

Proof: The proposed distributed algorithm is a modified Gauss-Seidel Algorithm which requires SBSs update the caching and routing policy sequentially. According to the Proposition 2,1 of Ch 3 in [26] in which gives if

(i) $f(y)$ is continuously differentiable and the constraint is convex and compact.
(ii) $f(y)$ is a unique minimizer when given the value of feasible y_{-n}.

For the first requirement, $f(y)$ is continuously differentiable. And we relax the integer variable to be continuous, so constraints are convex and compact.

$f(y)$ is a linear function which satisfies the second requirement. ■

This is a synchronized algorithm which requires all SBSs must update in one iteration. In practice, SBSs may not update in one iteration using possible outdated information. The asynchronized settings can be generalized by this algorithm while the convergence proof is more complex.

IV. Network Privacy Preserving Mechanism

In this section, we present the privacy preserving mechanism to protect the privacy of SBSs as well as MUs. The privacy of individuals information in the distributed communication network is significant which needs more concern. In many applications, SBSs belong to different companies. The caching policy and the routing policy are two key and private messages for each individual SBS. From the caching and routing policy, the private information of SBS and MUs can be deduced.

The caching policy is kept by SBS itself unless the SBS hacked by the attackers, and we do not consider this situation, if not, the caching policy can not be accessed by others. In our proposed distributed algorithm, the routing policy in each iteration is exchanged between each SBS and the BS, and the BS broadcasts aggregated routing policy to SBSs. This gives attackers opportunities to access the aggregated routing policy during the broadcasting. From the aggregated routing policy, the attacker can deduce the bandwidth condition of the SBS and the detailed sensitive information about MUs such as preference, location and so on. The individual privacy of SBSs and MUs will be compromised.

The traditional privacy methods put more attention on the channel encryption or add the not well-designed noise like random noise for individual nodes, e.g., for individual SBS or the set of MUs requests. The channel encryption focus on the physical layer of the transmission channel, we do not refer to this direction in this paper. Adding the noise directly overlooks the conditions that the attacker has the background information of the network. For example, when the attacker knows the topology structure of the network and a few individual MUs routing policies, the attacker can deduce precise information of other MUs or SBSs.

Here we face a dilemma: on one hand, the BS needs to broadcast the aggregated routing policy with SBSs, on the other hand, the privacy of the individual SBS and MUs should be protected. To tackle this problem, we design a privacy preserving mechanism based on differential privacy.

A. Background of Differential Privacy

Differential privacy [13] is a strong standard to preserve the privacy for the aggregate data with the privacy guarantee for algorithms. The basic idea of differential privacy is to add 'disturbance' for the result needs to be protected. The network processing privacy sensitive inputs from individuals is made deferentially private by adding a noise part for individual routing policy in such a way that the aggregated routing policy is not too sensitive to the data provided by any single SBS. As a result, it is provably difficult for an attacker, no matter how powerful, to make inferences about individual records from the published outputs. Differential privacy mechanism has been widely implemented in real systems in the industries like Google [15], and drawn great attention from the academe especially in machine learning [27]–[29]. Formally, the definition of differential privacy is as follows,

Definition 1. *Assume $\mathcal{D}$ and $\hat{\mathcal{D}}$ are neighboring database which differs by only one row. When a differential private algorithm $\mathcal{A}$ is implemented on data sets $\mathcal{D}$ and $\hat{\mathcal{D}}$, the output $\mathcal{A}(\mathcal{D})$ and $\mathcal{A}\left(\hat{\mathcal{D}}\right)$ do not change a lot.*

The following differential privacy should be met for any neighboring databases $\mathcal{D}$ and $\hat{\mathcal{D}}$, and for any $\mathcal{S} \in \mathcal{O}(outputrspace)$,

$$\frac{1}{e^{\epsilon}} Pr\left[\hat{\mathcal{A}}(\mathcal{D}) \in \mathcal{S}\right] \leq Pr\left[\mathcal{A}(\mathcal{D}) \in \mathcal{S}\right] \leq e^{\epsilon} Pr\left[\mathcal{A}(\mathcal{D}) \in \mathcal{S}\right]. \tag{26}$$

The sensitivity of the algorithm $\mathcal{A}$ is defined as: $\Delta\mathcal{A} = \max_{h(\mathcal{D},\hat{\mathcal{D}})=1} |\mathcal{A}(\mathcal{D}) - \hat{\mathcal{A}}((\mathcal{D})|$, where h denotes Hamming distance.

B. Laplace Privacy Preserving Mechanism (LPPM)

Several mechanisms are studied to preserve differential privacy such as the Laplace mechanism, the exponential mechanism and the Gaussian mechanism [13]. In this paper, we implement the Laplace mechanism to protect the privacy of the network and compute the sensitivity.

The commonly standard Laplace mechanism can not be applied to our network directly seeing that the complexity of our proposed model and the special structure of the distributed algorithm. We mainly face several challenges:

- The routing policy y_{nuf} belongs to $[0, 1]$. How to determine the Laplace component added to the routing policy considering the standard Laplace distribution (from $(-\infty, +\infty)$).

- After implementing the Laplace mechanism, the distributed algorithm converges or not.
- How to balance the trade-off between usefulness of the result and the privacy budget.

In Algorithm 1, the BS gathers individual routing policies from SBSs, and then broadcasts the aggregated routing policy to the SBS. To protect the privacy of individual SBS and MUs from the aggregated routing policy, each SBS adds the disturbance for the routing policy which follows a special form of the Laplace distribution. We denote the disturbance for the individual routing policy y_{nuf} as r_{nuf}. The new routing policy after adding the Laplace component is denoted as $\hat{y}_{nuf}$. The Laplace Distributed Privacy Preserving Mechanism (LPPM) is defined as follows.

Definition 2. *The Laplace Distributed Privacy Preserving Mechanism*

$$\hat{y}_{nuf} = y_{nuf} - r_{nuf}, \forall n \in \mathcal{N}, u \in \mathcal{U}. \tag{27}$$

Note that $y_{nuf} \in [0, 1]$, to adjust to this, we apply a special form of Laplace distribution for r_{nuf} [30]:

$$d_{nuf}(r_{nuf}) = \begin{cases} 0, & r_{nuf} \notin \mathcal{I}, \\ \frac{1}{\alpha}\frac{1}{2\beta}e^{-\frac{|r_{nuf}|}{\beta}}, & r_{nuf} \in \mathcal{I}, \end{cases} \tag{28}$$

where I=[0, $\delta \cdot y_{nuf}$]($\delta > 0$) is the interval of r_{nuf} where $\delta \in [0, 1)$ is the factor to adjust the impact of the r_{nuf} which can determine impact of the privacy. $\alpha(\beta) = \int_{\mathcal{I}} \frac{1}{2\beta} e^{-\frac{|r_{nuf}|}{b}} \mathrm{d}r_{nuf}$ is a normalisation parameter related with β. $\beta > 0$ is a given constant.

In LPPM, in each iteration τ of Algorithm 1, the positive disturbance r_{nuf} part is subtracted from the original routing policy y_{nuf}. This ensures that the request from the MU can be fully satisfied. From the objective function (6), if $r_{nuf} < 0$, then the $(1 - \sum_{n\in\mathcal{N}} \hat{y}_{nuf})$ will be greater than the $1 - \sum_{n\in\mathcal{N}} y_{nuf}$, this will cause that the total portion of the served content for the MU will be less than 1, and the request of the MU can not be fully served. When $r_{nuf} > 0$, the total served portion of content could be larger than 1, and the MU request can be fully satisfied. The extra part will be abandoned. In reality, the extra video packet will be discarded [31]. The total serving cost of the network will increase.

C. Convergence of the Distributed Algorithm

In this section, we address the convergence of the proposed distributed algorithm with the application of LPPM.

Theorem 3. *Given a fixed $\delta \in [0, 1)$ and $\epsilon \geq 0$, Algorithm 1 with the privacy mechanism LPPM converges,*

Proof: First, we prove that $\hat{y}_{nuf}$ has a fixed lower bound that $\hat{y}_{nuf}$ will not decrease to 0.

It is obvious to see from (27) that when given the fixed $\delta \in [0, 1)$, $\hat{y}_{nuf}$ has a lower bound that $\hat{y}_{nuf}$ will not decrease to 0. This prevents Algorithm 1 into an endless loop.

Next, we prove that in each iteration of Algorithm 1, $f(y)$ will decrease.

In each iteration τ, the caching policy x_{nf} is not disturbed by the Laplace noise, and it does not violate the constraint (2). We only consider the impact of routing policy y_{nuf}.

In the phase n of iteration of τ,

$$y_{nuf}(\tau) = \arg\min f(\hat{y}_{(i<n)uf}(\tau), y_{nuf}(\tau), \hat{y}_{(i>n)uf}(\tau - 1)), \tag{29}$$

then $\hat{y}_{nuf}$ is updated from (27). The new cost is renewed as $f(\hat{y}_{(i<n)uf}(\tau), \hat{y}_{nuf}(\tau), \hat{y}_{(i>n)uf}(\tau - 1))$. In the phase $\tau + 1$, the same updating is performed.

From (29), we obtain

$$f(\hat{y}_{(i<n)uf}(\tau), \hat{y}_{nuf}(\tau), \hat{y}_{(i>n)uf}(\tau - 1)) \geq$$
$$f(\hat{y}_{(i<n)uf}(\tau), \hat{y}_{nuf}(\tau), \hat{y}_{(n+1)uf}(\tau), \hat{y}_{(i>n+1)uf}(\tau - 1)),$$

for all updating. It indicates that the updating of the Algorithm1 is non-increasing. The rest of the proof is the same with the Chap. 3, the proposition 3.9 from [26]. ■

From Theorem 3, the Algorithm 1 still converges when subtracting a Laplace component for each routing policy in each iteration.

D. Performance Analysis

The proposed LPPM can preserve the privacy with the privacy guarantee of the Algorithm 1. However, it will affect the performance of network.

First, we state that LPPM we proposed can preserve the privacy of individual MUs.

Theorem 4. *Given a fixed $\delta \geq 0$ and $\epsilon \geq 0$, the LPPM is ϵ-differential privacy, if*

$$\beta \geq \frac{\Delta f(y)}{\epsilon} \tag{30}$$

The detailed proof will be provided in an expanded version of this paper. Inequality (30) is calculated according Eq.(2) of [30] directly. The proof is in a standard way like in [13], [30].

In LPPM, the core idea is subtracting the Laplace noise in a certain interval to the routing policy. On one side, this can help protect the privacy of individual MUs and SBSs, on the other side, the performance of the total network will be affected. We have the following theorem to show the quality of the LPPM.

Theorem 5. *Let x^* and y^* be the optimal caching and routing policy solved by Algorithm 1 without the LPPM. Accordingly, $f^*(y^*)$ is the optimal serving cost of the network. $\hat{x}$ and $\hat{y}$ denote the caching and routing policy with the implementation of the LPPM, $f(\hat{y})$ is the serving cost with the LPPM. The expected average of increase of cost satisfies that if $|y - \hat{y}| \leq \zeta$,*

$$E[f^*(y^*) - f(\hat{y})] \leq \Phi(\zeta)P_r + W(1 - P_r), \tag{31}$$

where $\Phi(\zeta)$ is a function of ζ, W is the maximum serving cost of the network, and P_r is the probability of $P(|y - \hat{y}| \leq \zeta)$.

Proof: First, we calculate the $Pr = P(|y-\hat{y}| \leq \zeta)$. From (27),

$$|y-\hat{y}| = \sum_{n\in\mathcal{N}}\sum_{u\in\mathcal{U}}\sum_{f\in\mathcal{F}} |r_{unf}| = \sum_{n\in\mathcal{N}}\sum_{u\in\mathcal{U}}\sum_{f\in\mathcal{F}} r_{nuf},$$

where r_{nuf} is the independent random variable. However, limited by the $|y-\hat{y}| \leq \zeta$, Pr can not be calculated directly by multiply each r_{nuf}'s probability. Instead, we derive the joint probability distribution $d(r)$ from the convolution of each r_{nuf}'s probability distribution:

$$d(r) = (d_{111} * \cdots * d_{nuf} * \cdots * d_{NUF})(r),$$

which can be calculated by the definition of evolution directly. Then,

$$P_r = \int_0^{\zeta} d(r).$$

When $|y-\hat{y}| \leq \zeta$,

$$|f^*(y^*) - f(y)| \leq f(\zeta) = \Phi(\zeta).$$

W is the maximum serving cost of the system, which indicates that the BS will serve all MUs requests directly, formally,

$$W = \sum_{u\in\mathcal{U}} \hat{d}_u \sum_{f\in\mathcal{F}} \lambda_{uf}.$$

From equations above, the expected cost increase of the network cost satisfies the following inequality,

$$E[f^*(y^*) - f(y] \leq \Phi(\zeta)P_r + W(1-P_r).$$

With LPPM, Algorithm 1 still guarantees the performance of the network. ■

V. Numerical Results

In this section, we evaluate the performance of the proposed distributed algorithm and the privacy preserving mechanism LPPM. We focus on evaluating the performance of the total serving cost for the whole communication network under different settings of base station capabilities. We aim to answer the following questions to understand the performance:

1) How much improvement of Algorithm 1 to reduce the cost compared with the existing scheme?
2) What is the impact of LPPM to the performance of the network?

In the simulation, we assume that there are three SBSs that can provide the MUs with the content cached on the edge server deployed. Other data contents requested are served by the BS from the core network. We assume that contents served by the BS cost much more than that served by SBSs in the edge. This model can be easily extended to multiple overlapped SBSs sets which can be viewed as various weight multiple models we propose above. However, it does not influence the performance of the proposed algorithm.

A. Simulation Setup

We implement the real data request trace from a well-known video stream website. We recorded the number of reviews of top 50 trending videos in 30 minutes on December 18, 2018 as the request number. Fig. 2 shows the number of reviews of first 20 videos. Some of videos are requested more than 140,000 time while some of them are only with a few thousands of requests. We further distributed requests randomly among MUs. Each request must be satisfied by SBSs and the BS, which combination must send the entire content to the user.

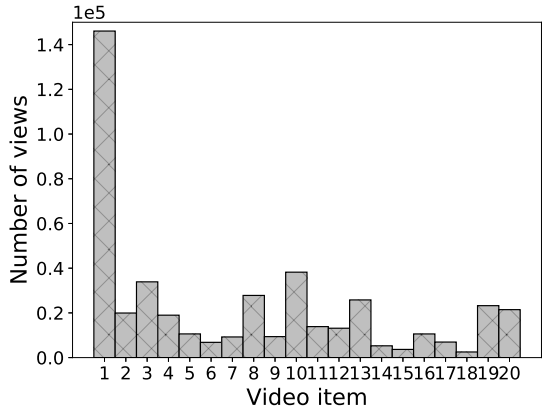

Fig. 2. A real request distribution got from the number of views of 20 videos in 30 minutes from a well-known streaming website.

Unless otherwise specify, we set the transmission efficiency parameter d_{nu} as 1 and the $\hat{d}_u$is randomly chosen between 100 and 150, since the BS is much farther away than SBSs to MUs.

We compare the performance of three schemes: the privacy preserving mechanism LPPM, the distributed algorithm (Algorithm 1) which is the optimal solution of the problem, and LRFU. LRFU is a classic caching replacement scheme which swaps the cached content based on the recent request frequency and time. We implement the problem using PuLP [32] and conduct the numerical simulation over a computer with Intel Core i5-4560 and 16GB RAM.

B. Performance over Different Privacy Budget

We first evaluate the total serving cost under different setting of privacy budget ϵ. We assume that there are 30 MUs requesting the aforementioned videos with the total 40 links between MUs and SBSs. The bandwidth of each SBS is set as 1000 unit at a time, and the bandwidth of BS is sufficient. The Laplace component factor δ is set as 0.5.

Fig. 3 demonstrates the impact of the budget privacy. It is worth noticing that optimum (Algorithm 1) and LRFU do not add the Laplace component, thus staying unchanged. Meanwhile, the cost of LPPM is decreasing as the ϵ increases. The larger ϵ means the smaller noise, which will make LPPM closer to the optimum. When $\epsilon = 0.01$, LPPM costs 10.1% more than optimum, and drops to only 1.2% when $\epsilon = 100$. This indicates that LPPM can achieve similar cost as the

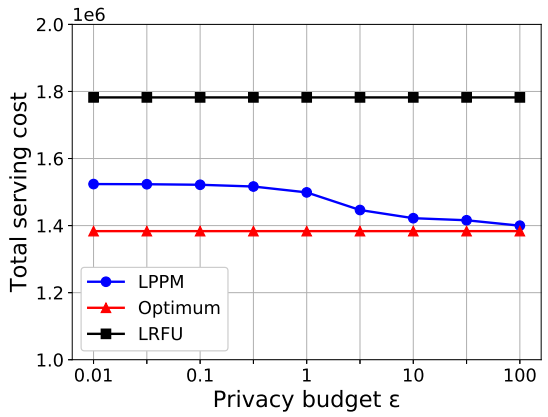

Fig. 3. The total serving cost under different settings of private budget. The higher budget means the lower interference, which our proposed mechanism will have a lower cost. Our proposed mechanism is very close to optimum results when the budget is high.

optimum results. Overall, our proposed mechanism is 17.3% better than LRFU in average, and only 6.6% more cost than the optimum.

C. Performance over Different Numbers of MUs

We then evaluate the total serving cost under different number of MUs. There are total 40 links among MUs and SBSs, and the bandwidth of each SBS is set as 1000. We set $\epsilon = 0.1$ at this scenario for the more obvious comparison.

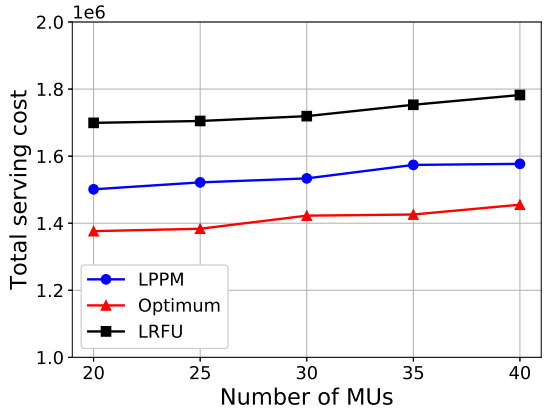

Fig. 4. The total serving cost under different number of MUs. More MUs bring more requests thus having more cost. Our proposed mechanism have tolerance under more MUs making requests.

Fig. 4 shows the change of the serving cost under different number of MUs. More MUs bring more requests which must be satisfied. Thus, the cost will increase when more MUs come and request data. Note that the increase rate is not significant. As for our proposed LPPM, there is only 5.1% increase in cost between 20 nodes and 40 nodes. Since popular data contents are often requested more, thus edge servers that cached such contents with free bandwidth can satisfy more nodes without too much cost imposed. Thus, the increasing is not significant. This also shows that our proposed system can tolerate with changes in number of MUs to some degree. Overall, our proposed mechanism is 11.0% better than LRFU in average, and 9.1% more cost than optimum.

D. Performance over Different Numbers of Links

Next, we evaluate the total serving cost under different number of links that MUs connect with SBSs. We set 30 MUs with variance total links to each SBS. The bandwidth capacity and $epsilon$ stay the same as previous problems.

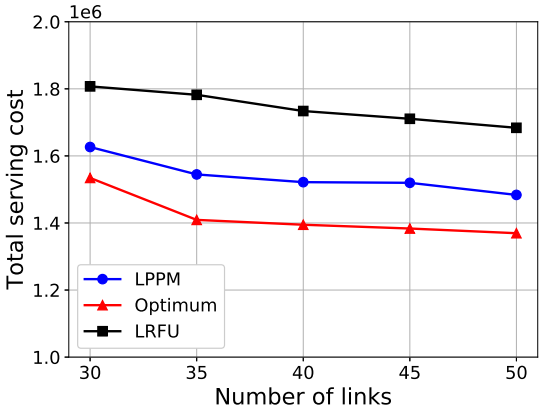

Fig. 5. The total serving cost under different number of links between MUs and SBSs. More links allows each SBS to serve more MUs, thus having lower cost. Our proposed mechanism performs well comparing with optimum under different numbers of links.

Fig. 5 shows the impact of total number of links over serving cost. The links difference in real world is due to the BS setups, environment blocks and so on. The more links exist, the better connection between MUs and SBSs will have. For each SBSs, it can serve more MUs; and for some MUs, they will have connections with more SBSs, from which they can receive more parts of data even any one of SBS cannot offer the entire content. The decreasing cost is not linear as the number of links, though. The cost decreases more when links are only a few, but stays most flat as larger number of links. Increasing links to some extent will have fewer impact due to the bottleneck like caches size, bandwidth capacity and other limitations. Overall, our proposed mechanism LPPM is cost 11.7% less than LRFU, and only 8.5% more than optimum.

E. Performance over Different Bandwidth

Finally, we evaluate how the bandwidth capacity of each SBS changes the total serving cost. We use in total 30 MUs and 40 links as previous simulations, and the privacy budget ϵ is set to 0.1.

Fig. 6 shows the total serving cost change over different bandwidth of SBSs. Since larger bandwidth means it can serve MUs more fraction of contents. Thus, MUs will require less from the BS for missing parts. We can notice the decrease of serving cost as bandwidth grows. The decreasing of serving cost is almost linear as the bandwidth is less than 1500

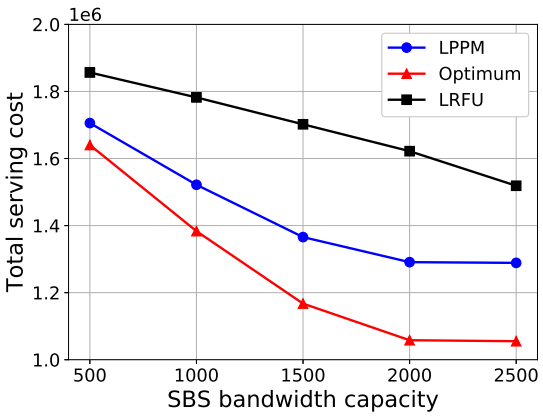

Fig. 6. The total serving cost under different bandwidth of SBSs. More bandwidth of a SBS can make it serve more MUs which will decrease the cost. Our proposed mechanism performs well comparing with the optimum under different bandwidth of SBSs.

units, then the decrease is slowing down. Similar to what we mentioned in the previous simulation, increase bandwidth to some extent will not have significant impact over the total serving cost, due to the limitations of other aspects. The LRFU through, has not reach such limits and still decreasing close to linearly. Our proposed mechanism LPPM still performs well under different conditions. Overall, our proposed mechanism DUA is cost 15.4% less than LRFU, and only 13.8% more than the optimum.

VI. Related Work

The studies about distributed edge computing or edge caching draw great attention with the development of the edge devices. Huge amount of work about distributed edge computing or caching are based on the distributed optimization. In [33], the authors consider a centralized caching model with cache replacement to minimize the total serving cost. Liu *et al.* [3] proposes a simple but representative distributed caching model to minimize the average delay for MUs. The proposed algorithm is based on the Belief Propagation. Messages sharing among various nodes helps to approach the global optimum. Belief Propagation is also applied in [4].

The majority of distributed caching papers model the caching problem as the linear or convex optimization which consider various model to improve the performance of the network. Some paper model the caching policy as a continuous variable which can guarantee the convexity of the optimization. In [34], a femto-caching scheme is proposed for a cellular network combined with SBSs to reduce the transmission delay imposed. The caching variable is relaxed to be continuous make the computational comlexity lower. In [5], [6], [35]–[37], the caching variable is modeled as the 0-1 integer variable. In [36], authors consider two models, Maximum Ratio Transmission (MRT) and Zero-Forcing Beamforming (ZFBF), and maximize the successful transmission rate to satisfied the MUs' requirement.The near-optimal strategies of caching is derived by relaxing the integer variable. A more specific case is discussed in [37], the authors consider a distributed model which cached files are videos coded by Salable Video Coding. It is more complicated with the constraint of the video coding. Authors in [35] consider a tree-structure cache station model, and develop a cooperative cache management algorithms to maximize the traffic volume served from cache and minimizing the bandwidth cost. A dynamic model is presented in [6], a distributed bounded approximation algorithm is proposed to fit for various applications in the edge. In [5], authors consider two factors the routing latency and network failures in a across a resilient cache network. A distributed gradient ascent algorithm with the performance guarantee is proposed to minimize the transmission latency. The paper mentioned above is more focus on the specific case, which can not represent the fundamental situation in the caching.

With the fast development of Artificial intelligence (AI), leaning techniques are implemented in the edge. In [7], [38], the authors implement the reinforcement learning to determine the optimal caching policy. In [39], authors propose a gradient descent based approaches to learn model parameters from data distributed across multiple edge nodes. The privacy of the users can be preserved without sending data to a centralized place. However, the performance can not be guaranteed by the learning method.

Differential privacy [13], [14] can constitute the privacy guarantee for gathered data in the implementation of algorithms. Differential privacy has been widely used in real system to protect the privacy of the individual users [15]. In distributed learning, differential privacy are widely studied to protect the privacy of learning in the gradient decent. [27], [29] study the performance guarantee on Deep Learning and Federated Learning. Various learning tasks are conducted in [40]. However, as a promising privacy preserving technique, differential privacy is seldom considered or applied in the edge network in which the security and privacy have significant influence on the total network.

VII. Concluding Remarks

In this paper, we investigate how to preserve the privacy of mobile users and independent small base stations in the distributed edge network with the privacy guarantee by proposing a mechanism LPPM (Laplace Preserving Privacy Mechanism). We first present a synchronized distributed algorithm to determine the optimal caching and routing policy to minimize the total serving cost of the network. Then, we introduce LPPM and prove the privacy guarantee. Numerical evaluation based on the real-world trace highlights the efficiency of the proposed distributed algorithm and LPPM. In the future, we plan to discuss the performance of the asynchronized algorithm and other privacy preserving mechanisms.

Acknowledgment

This work is supported in part by US National Science Foundation under grant numbers 1513719 and 1730291.

REFERENCES

[1] J. G. Andrews, S. Buzzi, W. Choi, S. V. Hanly, A. E. Lozano, A. C. K. Soong, and J. Zhang, "What will 5G be?" *IEEE Journal on Selected Areas in Communications*, vol. 32, pp. 1065–1082, 2014.

[2] F. Liu, G. Tang, Y. Li, Z. Cai, X. Zhang, and T. Zhou, "A survey on edge computing systems and tools," *Proceedings of the IEEE*, vol. 107, no. 8, pp. 1537–1562, 2019.

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

[4] J. Li, Y. Chen, Z. Lin, W. Chen, B. Vucetic, and L. Hanzo, "Distributed caching for data dissemination in the downlink of heterogeneous networks," *IEEE Transactions on communications*, vol. 63, no. 10, pp. 3553–3568, 2015.

[5] J. Li, T. K. Phan, W. K. Chai, D. Tuncer, G. Pavlou, D. Griffin, and M. Rio, "DR-Cache: Distributed resilient caching with latency guarantees," in *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*. IEEE, 2018, pp. 441–449.

[6] G. Castellano, F. Esposito, and F. Risso, "A distributed orchestration algorithm for edge computing resources with guarantees," in *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*. IEEE, 2019, pp. 2548–2556.

[7] L. Lu, Y. Jiang, M. Bennis, Z. Ding, F.-C. Zheng, and X. You, "Distributed edge caching via reinforcement learning in fog radio access networks," in *2019 IEEE 89th Vehicular Technology Conference (VTC2019-Spring)*. IEEE, 2019, pp. 1–6.

[8] A. Sadeghi, F. Sheikholeslami, and G. B. Giannakis, "Optimal and scalable caching for 5G using reinforcement learning of space-time popularities," *IEEE Journal of Selected Topics in Signal Processing*, vol. 12, no. 1, pp. 180–190, 2017.

[9] P. Garcia Lopez, A. Montresor, D. Epema, A. Datta, T. Higashino, A. Iamnitchi, M. Barcellos, P. Felber, and E. Riviere, "Edge-centric computing: Vision and challenges," *ACM SIGCOMM Computer Communication Review*, vol. 45, no. 5, pp. 37–42, 2015.

[10] W. Shi and S. Dustdar, "The promise of edge computing," *Computer*, vol. 49, no. 5, pp. 78–81, 2016.

[11] J. Zhao, R. Mortier, J. Crowcroft, and L. Wang, "Privacy-preserving machine learning based data analytics on edge devices," in *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*. ACM, 2018, pp. 341–346.

[12] S. R. Hussain, M. Echeverria, I. Karim, O. Chowdhury, and E. Bertino, "5GReasoner: A property-directed security and privacy analysis framework for 5g cellular network protocol," in *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, 2019, pp. 669–684.

[13] C. Dwork, A. Roth *et al.*, "The algorithmic foundations of differential privacy," *Foundations and Trends® in Theoretical Computer Science*, vol. 9, no. 3–4, pp. 211–407, 2014.

[14] C. Dwork, "A firm foundation for private data analysis," *Communications of the ACM*, vol. 54, no. 1, pp. 86–95, 2011.

[15] C. Dwork and F. D. McSherry, "Differential data privacy," Apr. 13 2010, uS Patent 7,698,250.

[16] Y. Zhao, J. Zhao, L. Jiang, R. Tan, and D. Niyato, "Mobile edge computing, blockchain and reputation-based crowdsourcing iot federated learning: A secure, decentralized and privacy-preserving system," *arXiv preprint arXiv:1906.10893*, 2019.

[17] V. Sharma, I. You, D. N. K. Jayakody, and M. Atiquzzaman, "Cooperative trust relaying and privacy preservation via edge-crowdsourcing in social internet of things," *Future Generation Computer Systems*, vol. 92, pp. 758–776, 2019.

[18] X. Li, J. Yang, Z. Sun, and J. Zhang, "Differential privacy for edge weights in social networks," *Security and Communication Networks*, vol. 2017, 2017.

[19] M. Du, K. Wang, Y. Chen, X. Wang, and Y. Sun, "Big data privacy preserving in multi-access edge computing for heterogeneous internet of things," *IEEE Communications Magazine*, vol. 56, no. 8, pp. 62–67, 2018.

[20] R. Pedarsani, M. A. Maddah-Ali, and U. Niesen, "Online coded caching," *IEEE/ACM Transactions on Networking*, vol. 24, no. 2, pp. 836–845, 2015.

[21] K. Poularakis, G. Iosifidis, A. Argyriou, I. Koutsopoulos, and L. Tassiulas, "Caching and operator cooperation policies for layered video content delivery," in *IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on Computer Communications*. IEEE, 2016, pp. 1–9.

[22] K. Poularakis, G. Iosifidis, I. Pefkianakis, L. Tassiulas, and M. May, "Mobile data offloading through caching in residential 802.11 wireless networks," *IEEE Transactions on Network and Service Management*, vol. 13, no. 1, pp. 71–84, 2016.

[23] V. K. Adhikari, Y. Guo, F. Hao, V. Hilt, Z.-L. Zhang, M. Varvello, and M. Steiner, "Measurement study of netflix, hulu, and a tale of three cdns," *IEEE/ACM Transactions on Networking*, vol. 23, no. 6, pp. 1984–1997, 2014.

[24] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

[25] D. P. Bertsekas and A. Scientific, *Convex Optimization Algorithms*. Athena Scientific Belmont, 2015.

[26] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*. Englewood Cliffs, NJ: Prentice Hall, 1989, vol. 23.

[27] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, "Deep learning with differential privacy," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2016, pp. 308–318.

[28] A. Friedman and A. Schuster, "Data mining with differential privacy," in *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2010, pp. 493–502.

[29] R. C. Geyer, T. Klein, and M. Nabi, "Differentially private federated learning: A client level perspective," *arXiv preprint arXiv:1712.07557*, 2017.

[30] N. Holohan, S. Antonatos, S. Braghin, and P. Mac Aonghusa, "The bounded laplace mechanism in differential privacy," *arXiv preprint arXiv:1808.10410*, 2018.

[31] H. Schwarz, D. Marpe, and T. Wiegand, "Overview of the scalable video coding extension of the h. 264/avc standard," *To appear in IEEE Transactions on Circuits and Systems for Video Technology*, p. 1, 2007.

[32] S. Mitchell, M. OSullivan, and I. Dunning, "Pulp: a linear programming toolkit for python," *The University of Auckland, Auckland, New Zealand*, 2011.

[33] Y. Zeng, Y. Huang, Z. Liu, and Y. Yang, "Joint online edge caching and load balancing for mobile data offloading in 5g networks," in *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 2019, pp. 923–933.

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

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

[36] W. C. Ao and K. Psounis, "Fast content delivery via distributed caching and small cell cooperation," *IEEE Transactions on Mobile Computing*, vol. 17, no. 5, pp. 1048–1061, 2018.

[37] K. Poularakis, G. Iosifidis, A. Argyriou, I. Koutsopoulos, and L. Tassiulas, "Distributed caching algorithms in the realm of layered video streaming," *IEEE Transactions on Mobile Computing*, vol. 18, no. 4, pp. 757–770, 2019.

[38] L. Marini, J. Li, and Y. Li, "Distributed caching based on decentralized learning automata," in *2015 IEEE International Conference on Communications (ICC)*. IEEE, 2015, pp. 3807–3812.

[39] S. Wang, T. Tuor, T. Salonidis, K. K. Leung, C. Makaya, T. He, and K. Chan, "When edge meets learning: Adaptive control for resource-constrained distributed machine learning," in *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*. IEEE, 2018, pp. 63–71.

[40] E. Bagdasaryan, O. Poursaeed, and V. Shmatikov, "Differential privacy has disparate impact on model accuracy," in *Advances in Neural Information Processing Systems*, 2019, pp. 15 453–15 462.