Optimal and Practical Algorithms for Implementing Wireless CDN Based on Base Stations

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*—The development of Network Function Virtualization (NFV) and Software Defined Networks (SDN) standards is an opportunity for Mobile Network Operators (MNOs) to deploy Content Delivery Network (CDN) functionalities into the mobile network edge, such as Base Stations (BSs). In this paper, we investigated the content placement problem for the BS-based wireless CDN system. We call the storage resources implemented on BSs as *storage helpers*. Due to the limited helper storage capacity and the limited user population served per BS, helpers exhibit low hit ratio comparing to traditional CDN edge servers serving a wide area. Then, cooperation is a suitable means to enhance the performance of the wireless CDN system. We propose that BSs close to each other cooperate in replicating content and replying user requests. We formulate the optimum content placement problem to minimize the traffic pressure on mobile network gateways, and show the problem complexity is NP-Hard. We then transform the problem into a *multiple-Maximum Weighted Independent Set* problem, and propose a heuristic algorithm. The evaluation shows that the hit ratio is improved by our algorithm comparing to the traditional Least Frequently Used (LFU) policy without cooperation.**

I. INTRODUCTION

The development of Network Function Virtualization (NFV) and Software Defined Networks (SDN) standards is an opportunity for the Mobile Network Operators (MNOs) to propose new services to content providers (CPs) [1]. The deployment of Content Delivery Network (CDN) deep into the mobile network infrastructure gets a growing attention, because it can fulfil the demand of both the CPs and the MNOs. For the CPs, the availability of storage resources near the end-users is the promise of a better Quality of Experience (QoE). Since CPs prefer managing storage resources by themselves (for better control of video quality, ad insertion and log management), the combination of easy-to-use, standardized NFV storage resources near the end-users is an attractive service. For the MNOs, renting edge storage resources is a way to alleviate the formidable traffic burden on the mobile gateway and backhaul links. The recent adoption of secured protocols (HTTPS) by most of the CPs has significantly reduced the performances of caches [2]. By implementing middleboxes that are partly controlled by third parties, the MNOs have a chance to diminish the traffic on their core network.

In the scope of wired network, strategies have been developed to take into account the presence of manageable CDN servers directly in the network of Internet Service Providers (ISPs), which is named as Telco-CDN (e.g., [3]). Whereas for wireless network, the literature on this area has stayed remarkably low. The recent ETSI Mobile Edge Computing (MEC) group also aim to offer CPs with storage and computing capabilities at the edge of the mobile network, which confirms our inspiration of implementing *storage helpers* in Base Stations (BSs) to establish BS-based wireless CDNs.

Because the BS helpers have limited storage capacity, only the most popular resources can be stored. Moreover, nearby BSs have high chances to store the same content, which leads to inefficiencies when see from a broader perspective. An idea, which has been studied for a long time in the context of caches [4], is to enable cooperation between closing helpers. The cooperation between storage-featuring BSs especially suits the architecture of modern cellular networks. When a request misses the content on the BS i which the user is attached to, the request is redirected to a nearby BS j, which is in cooperation with BS i. Three different technologies allow data transmission between BSs: (i) via the backhaul link [5], (ii) with the Local IP Access and Selected IP Traffic Offload (LIPA/SIPTO) technique [6], and (iii) by exploring the usage of the X2 interface to enable direct data transmission between BSs [7]. For these three techniques to be efficient, BSs should be in cooperation with a small number of other BSs.

In this paper, we study the scenario where each BS is equipped with storage helper and each BS is allowed to cooperate with other BSs in a given range. We investigate the optimal content placement for the content provider in such a cooperative BS-based CDN. We formulate the content placement problem to minimize the throughput on mobile gateway and show that the problem complexity is NP-Hard. In order to solve the problem, we transform it into a multiple-Maximum Weighted Independent Set (MWIS) problem. The MWIS problem is known as hard to approximate, we then design a heuristic algorithm and discuss some practical implementation relevance. We compare the performances of our heuristic to a resource management policy that is inspired by the Least Frequently Used (LFU) policy in cache systems. The evaluation confirms that the hit ratio of our cooperative BS-based CDN outperforms this traditional approach.

II. RELATED WORKS

In the literature, works related to implementing CDN functionalities into the mobile network infrastructure stays

---

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

978-1-5090-1698-3/16/$31.00 ©2016 IEEE

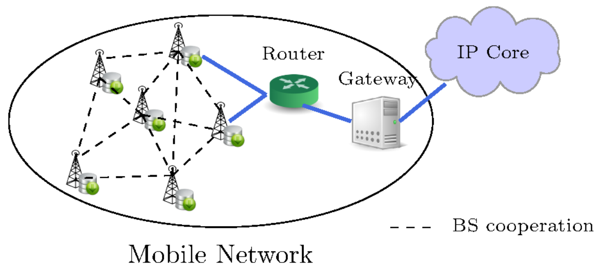

Fig. 1: The cooperative BS-based wireless CDN system

remarkably low. In [8], the authors looked to deploy CDN mechanisms onto mobile platforms by exploring storage capacities of mobile devices, and studied the corresponding content replication problem. In [9], the authors investigated the energy efficiency aspect of a CDN system based on Mesh WiFi implementation. None of these works consider storage helpers on BSs.

Content placement is one of the key issues for cache based system, thus is intensively studied in the background of caching. In [10], the authors propose reactive caching policies for BS caching system to reduce traffic throughput on backhaul links. In [11], the authors propose to implement caches on small cells as storage helpers for mitigating traffic pressure on backhaul links. They studied the optimum content placement problem to minimize the file downloading time. In [12], the authors design algorithm 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 [5], the authors investigate content replacement strategy on BS cache. The problem is modeled by Markov decision process, and a distributed content placement algorithm is proposed.

In the above related works, BSs are allowed to cooperate within the range of the whole mobile network. However, the mobile network is a large scale system which involves a huge number of BSs. Cooperating across the whole network leads to complexity in content discovery, user request redirection and load balancing. Hence, in the current work, we restrict each BS helper to cooperate within a certain range in order to make the mobile network architecture more flat [7].

III. SYSTEM MODEL AND PROBLEM FORMULATION

A. The Cooperative BS-based CDN System

Mobile BSs are equipped with storage helpers to serve mobile users' content requests. In the following presentation, the two terms, storage helper and BS, are used interchangeably. We consider a region in which N BSs geographically deployed and served by one gateway, thus BSs are indexed by $n = 1, \cdots, N$. Each BS is equipped with a helper of size C_n.

Mobile users are associated to their local BS. When a user request is missing, the request is redirected and replied by another BS helper which is in cooperation with the local one. We define a neighbor BS set for each BS, among which cooperation is allowed. We denote by $N(n)$ the open neighbor set without including BS n, and $N[n]$ the close neighbor set with n respectively. To determine $N(n)$, the MNO measures some statistical metrics. In the current paper, we consider the transmission delay such that the neighbor set $N(n)$ is the set of BSs that have short delay to BS n. A similar approach has been used in [13] for wired cooperative CDN. Formally, we could define the (open) neighbor set of n as: $N(n) = \{n' | d(n, n') \leq D_T\}$, where $d(n, n')$ measures the statistical transmission delay from n' to n, and D_T is a given delay threshold. Moreover, for simplicity, we assume symmetric neighborhood relationship, which means that $n' \in N(n)$ implies $n \in N(n')$.

Please note that the range of $N(n)$ should be properly defined by considering multiple aspects. A too large cooperation range, for example the whole network, increases complexity in implementation and content discovery. A too small range leads to inefficient use of storage capacity. Besides, user request density should also be considered for load balancing. However, in the current work, we assume that $N(n)$ is given.

The BS-based wireless CDN system is sketched in Fig. 1. The dashed lines between BSs show the cooperation relationship. Based on the mobile network topology and the BSs cooperation relationship, we define a graph $G = (\mathcal{V}, \mathcal{E})$ to represent the system. The set $\mathcal{V}$ represents the BSs (helpers). The set $\mathcal{E}$ shows BSs cooperation such that a link $(n, n') \in \mathcal{E}$ if $n \in N(n')$.

The aim of MNOs is to minimize traffic transmitted through the mobile gateway to lower their operational cost. In this architecture, a vast number of BSs are involved because one gateway serves a geographically wide area with many BSs. In the above process, the challenge is how to determine the optimum content placement for the MNO's interest. In the following, we will formulate the problem.

B. Content Placement Problem Formulation

Suppose the CP aims to store a set of M videos, and we denote each video by $m = 1, \cdots, M$. Let s_m denotes the size of video m. There are U users to be served in the system, and users are indexed by $u = 1, \cdots, U$. Proactive content placement is based on accurate user request prediction. Hence, we denote by p_u^m the probability that user u requests content m, and assume that this information is known by CP. Besides, we have $\sum_m p_u^m = 1, \forall u$.

Mobile users handover among BSs as they move. The time-homogeneous discrete-time Markov chain (DTMC) is proper to model an individual user's mobility. The state space is $S = \{S_1, S_2, \ldots, S_N\}$, with S_n representing that the user is currently accessing to BS n. Then, the state transition represents user handover from one BS to another. Let $pt_{nn'}^u$ denotes the transition probability of user u moving from BS n to n'. The transition occurs between neighboring BSs (please note that this neighbor is different from the neighbor set defined above). Then DTMC model can be learned out of the trail of mobility traces of an individual user [14]. Assuming that the DTMC is ergodic, which means that all states are aperiodic and positive recurrent. Then, we can derive the steady-state transition probability, which is the long-term probability of user u accesses to BS n, noted by p_u^n. Thus, $p_u^n = \sum_{n'=1}^{N} p_u^{n'} pt^u_{n'n}, \forall n, u$, and $\sum_{n=1}^{N} p_u^n = 1, \forall u$.

We define a binary variable x_n^m such that $x_n^m = 1$ means BS n stores the video m, and 0 otherwise. For the purpose of storage space saving, under proper cooperation range, one replica of the video stored in one neighborhood is sufficient to fulfill the video demand from all the involved BSs, thus we have: $\sum_{n' \in N[n]} x_{n'}^m \leq 1, \forall n, m$.

Given a content placement $\mathbf{X}$, the probability that a user u's video request can be replied by the BS CDN system is:

$$\sum_m \sum_n p_u^n p_u^m \sum_{n' \in N[n]} x_{n'}^m \qquad \forall u.$$

Our objective is to minimize the throughput on the gateway. The expectation of traffic incurred by user u on the gateway (T_u) is the expectation of traffic demanded by the user minus the expectation of traffic saving by using BS storage helpers:

$$E\{T_u\} = \sum_m p_u^m s_m - \sum_m \sum_n s_m p_u^n p_u^m \sum_{n' \in N[n]} x_{n'}^m, \qquad \forall u.$$

Thus, the expectation of overall traffic consumption on gateway is expressed as:

$$\begin{aligned} E\{T_{all}\} &= \sum_u E\{T_u\} \\ &= \sum_u \sum_m p_u^m s_m - \sum_u \sum_m \sum_n s_m p_u^n p_u^m \sum_{n' \in N[n]} x_{n'}^m. \end{aligned}$$

Our objective is to minimize $E\{T_{all}\}$. This objective is equivalent to the following one:

$$\textbf{maximize} \quad \sum_u \sum_m \sum_n s_m p_u^n p_u^m \sum_{n' \in N[n]} x_{n'}^m \qquad (1)$$

s.t.:

$$\sum_m s_m x_n^m \leq C_n \qquad \forall n, \qquad (2)$$

$$\sum_{n' \in N[n]} x_{n'}^m \leq 1, \qquad \forall n, m, \qquad (3)$$

$$x_n^m \in \{0, 1\}, \qquad \forall n, m \qquad (4)$$

where constraint (2) is the storage capacity constraint and constraint (3) is the cooperation constraint.

IV. PROBLEM ANALYSIS

In this problem formulation, we aim to minimize the traffic throughput on mobile network gateway, and do not consider the transmission cost from gateway and neighbor BSs to the local BS. In our BSs cooperation mechanism, the transmission cost can be regarded as a simplified binary cost model: (1) no transmission cost between neighboring BSs; (2) large cost (as from the gateway) between non-cooperative BSs. Thus, storing the content locally or on neighbors is regarded as the same. In the following, we analyze the cooperating content placement problem in complexity and propose solutions.

Observation 1. *The complexity of the above problem formulation is NP-Hard.*

By enlarging the neighbor set of each BS to all BSs, the problem becomes the *Generalized Assignment Problem* (GAP) which belongs to the *Knapsack problem* family. The GAP is NP-Hard, thus the complexity of our problem is at least NP-Hard. As a result, it follows that, unless P=NP, the problem admits no polynomial-time exact algorithm.

Observation 2. *For each video file m, the BSs storing the file form a $(2,1)$-Independent Set in the graph G.*

From constraint (3), a neighborhood stores at most one copy of each video. This implies that when a BS stores a given video, it cannot find another BS storing the same video within two hops, otherwise, the BSs laying between violates constraint (3). In other words, in the graph G, a selected helper node for m is adjacent to no other helper node for the same video within 2 hops. This indicates that a feasible solution $\bar{x}^m$ is a $(2,1)$-*Independent Set* of graph G, $\forall m$.

The $(2,1)$-Independent Set of graph G corresponds to the Independent Set (IS) of the graph G^2, where G^2 is the second power of G. We then reformulate the original problem as follows:

Transformed Integer Linear Program Formulation

$$\textbf{max.} \quad \sum_n \sum_m d_n^m x_n^m \qquad (5)$$

$$\textbf{s.t.} \quad d_n^m = \sum_{n' \in N[n]} \sum_u s_m p_u^{n'} p_u^m, \qquad \forall n, m \quad (6)$$

$$\sum_m s_m x_n^m \leq C_n, \qquad \forall m \quad (7)$$

$$x_n^m + x_{n'}^m \leq 1, \qquad \forall m, (n, n') \in \mathcal{E}(G^2) \quad (8)$$

$$x_n^m \in \{0, 1\}, \qquad \forall n, m \quad (9)$$

The objective (5) comes from the fact that the aggregate demands of all neighbors of BS n are served by videos stored on n. Equation (6) calculates the aggregate demands of all neighbors of BS n for video m (noted as d_n^m). Constraint (7) shows storage capacity limitation and constraint (8) restricts IS on G^2 for each video. The notation $\mathcal{E}(G^2)$ represents the edge set of graph G^2.

This formulation is closely related to the *Maximum Weighted Independent Set* (MWIS) problem, which is also known as NP-Hard and hard to approximate [15]. Our problem is even more complex in the sense that it should consider multiple videos and the storage capacity constraint. For small scale network, it is possible to calculate the optimum solution using tools such as CPLEX. For realistic large network, we design a heuristic algorithm to return a fast result.

V. HEURISTIC ALGORITHM AND PRACTICAL RELEVANCE

A. Algorithm

Without loss of generality, we assume that videos have uniform file size s, and helpers have uniform storage capacity.

Then, we denote by C the number of videos a helper can store as: $C = \lfloor \frac{C_n}{s} \rfloor$; and $C_r(n)$ the spare capacity of BS n.

The aggregate video demand d_n^m of BS n on video m can be regarded as the *weight* of BS n for video m. In the initialization phase, we denote by d^{m*} the highest weight of all helpers for video m. Then, for all videos $\{m_i | 1 \leq i \leq M\}$, we sort the highest weight in decreasing order in $list_M = \{m_1, \cdots, m_M\}$, such that $d^{m_1*} \geq \cdots \geq d^{m_i*} \geq \cdots \geq d^{m_M*}$. Then, for each video m_i, we sort the weight of all helpers in decreasing order: $list_{m_i} = \{d^{m_i}_{n_{i1}}, \cdots, d^{m_i}_{n_{ij}}, \cdots, d^{m_i}_{n_{iN}}\}$ such that $d^{m_i}_{n_{i1}} \geq \cdots \geq d^{m_i}_{n_{ij}} \geq \cdots \geq d^{m_i}_{n_{iN}}$. Please note that, by this definition, $d^{m_i}_{n_{i1}} = d^{m_i*}$.

The idea behind our algorithm is that we start by finding an IS with sufficient large weights from the video with the highest d^{m*}, for instance m_1 (line 9). We note by n^* the index of helper for m_1 such that $d^{m_1}_{n^*} = min\{d^{m_1}_n \geq \frac{d^{m_2*}}{\alpha} | \forall n, \alpha > 1\}$. Then, helpers with weight higher than $d^{m_1}_{n^*}$ will be put into a set A_{m_1}, to which a MWIS algorithm is applied (line 10-12). Then, the selected helpers are put into set B_{m_1}, which contains all BSs that replicate m_1. The idea of defining $d^{m_1}_{n^*}$ is that when the weight of helpers decrease to a level that is considered as sufficient lower than that of the next video m_2 ($< \frac{d^{m_2*}}{\alpha}$), then costing one capacity to store m_1 to gain some low weight is no better that considering the next file m_2.

For m_1, it updates $list_{m_1}$ by deleting helpers in $\{n | (n, n') \in \mathcal{E}(G^2), n' \in B_{m_1}\}$ (line 14). Then, m_1 is inserted in $list_M$ in a correct position with the updated highest weight (line 16). The selected helpers also updates their remaining capacity. Then, we consider m_2, which is currently the video with highest weight. The sets A_{m_2} and B_{m_2} are defined in the same way. Thus, our algorithm always consider the video with the currently highest weight, and select MWIS until the weight of the current video drops into a significant low level, where the significance is set by the parameter α.

In the above process, each step calculates an IS for a given set of BSs, and subtracts the resulting IS and all its neighbors in G^2. Thus, the resulting storage for each video also forms an IS. The pseudo-code of the algorithm is given in Algorithm 1.

Algorithm 1 Heuristic

```
1: Input: $G^2$, $\{d_n^m\}$, $\alpha$
2: Output: $\{B_{m_i}\}, 1 \leq i \leq M$
3: Initialization:
4: $list_M = \{m_i\}$ in $d^{m_i*}$ decreasing order, $1 \leq i \leq M$
5: $list_{m_i} = \{d_n^{m_i}\}$ in $d_n^{m_i}$ decreasing order, $\forall n$, $1 \leq i \leq M$
6: $B_{m_i} = \emptyset, 1 \leq i \leq M$ and $C_r(n) = C, \forall n$
7: Main part:
8: while $list_M \neq \emptyset$ and $\exists n$ s.t. $C_r(n) > 0$ do
9:     $m_i = \leftarrow$ POP $list_M$
10:    $D = d^{m_{i+1}*}$
11:    $A_{m_i} = \{n | \forall n, \text{ s.t. } d_n^{m_i} \geq \frac{D}{\alpha}, C_r(n) > 0\}$
12:    $B_{m_i} = B_{m_i} + MWIS(A_i)$
13:    update $n \in B_{m_i}$ their storage state
14:    update $list_{m_i}$ by deleting $n \in \{n | (n, n') \in \mathcal{E}(G^2), n' \in B_{m_1}\}$ and $n$ with $C_r(n) = 0$
15:    if $list_{m_i} \neq \emptyset$ then
16:        update $list_M$ by inserting $m_i$ in right position
17:    end if
18: end while
```

Algorithm 2 Practical Improvement on BS n

```
Initialization:
$list_{rm} = \{m | n$ does not store $m\}$
for $m \in list_{rm}$ calculate $r_n^m = \sum_u s_m p_u^m p_u^n$
sort $list_{rm}$ in $r_n^m$ decreasing order
Main part:
while $list_{rm} \neq \emptyset$ and $C_r(n) > 0$ do
    $m = \leftarrow$ POP $list_{rm}$
    store $m$ and update $C_r(n)$
end while
```

B. Practical Relevance

Practical Improvement. Please note that the physical meaning of our problem objective formulation (throughput on gateway) is based on the constraint (3), which restricts the number of copies within one neighborhood. The constraint (3) is proposed for storage space saving. However, it can happen that some constraints in (2) are not actively tight, which means that some helpers have un-allocated storage spaces. In this case, we propose a practical improvement mechanism (Algorithm 2) to fully utilize the storage capacities and further enhance system performance. We define $r_n^m = \sum_u s_m p_u^m p_u^n$ to represent local video demand on BS n for video m. Then, helpers with spare capacity sort $\{r_n^m\}$ in decreasing order, and stores the content which is not yet stored.

Request Routing. Our cooperation mechanism facilitates content discovery by restricting request redirection within the neighbor range. The content placement algorithm can be executed on a central entity of the network. Then, the storage decision and content redirection information are disseminated through messages, from which each helper can build a content routing table includes all storage information of its neighbors. By maintaining such a routing table, each helper could effectively route user requests. For content which is not stored by any helper in the neighborhood, it will be directly retrieved from the original server through the mobile gateway.

VI. EVALUATIONS

We build a simulation platform to evaluate the performance of our proposed heuristic content placement algorithm. In order to emulate a realistic network, our simulation includes $1,200$ BSs served with one gateway, as experimented in [16]. Each BS randomly chooses a number of BSs as neighbors. We aim to disseminate 10 categories of videos. Each category contains 20 videos. That is 200 videos in total. The content popularity follows zipf's law within each category, and users request videos based on content popularity. We assume users access each BS with uniform probability. The storage size is 100 GB for all BS helpers, and each video needs a storage space of 5 GB. We vary the number of users served per BS

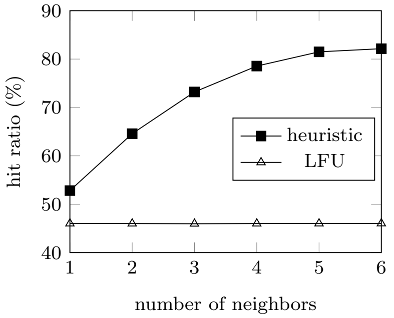

Fig. 2: Average hit ratio for 100 users per BS

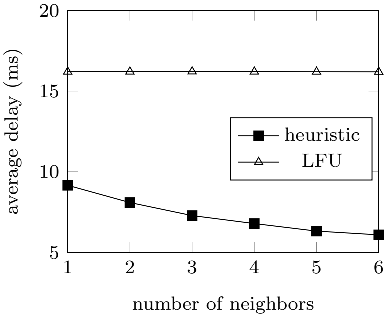

Fig. 3: Average delay for 100 users per BS

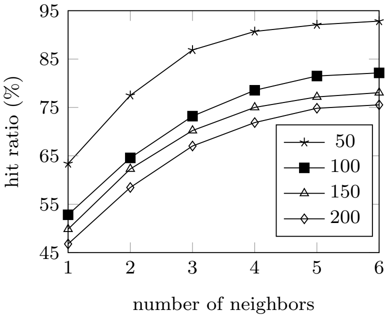

Fig. 4: Hit ratio for different number of users per BS

from 50 to 200. The α parameter used in the heuristic is set to 2.

We define the percentage of user requests replied by helpers (either from the local or from the neighboring ones) as the hit ratio. A higher hit ratio corresponds to lower throughput on mobile network gateway. We first fix the number of users per BS to 100, and vary the number of neighbor BSs from 1 to 6. In Fig. 2, we show the hit ratio of our algorithm, and compare to that of the Least Frequently Used (LFU) content caching policy. For LFU, it can achieve a hit ratio of about 46%. For our heuristic, when the number of neighboring BSs increases, the hit ratio increases from 52% for one neighbor to 82% for six neighbors. We also observe that the hit ratio exhibits diminishing additional benefits for larger cooperation range. This indicates a proper $N(n)$ value should be defined for trading off between performance and complexity.

In Fig. 3, we show the average video transmission delay for 100 users per BS. We set the transmission delay from local BS helper as 0 ms, from neighbor BS as 10 ms, and from the gateway as 30 ms. The results are coherent to that of the Fig. 2. In Fig. 4, we show the hit ratio for different number of users per BS. We increase user population per BS from 50 to 200, which represents different levels of requirement density. For 50 users, the system endures light requirement, the hit ratio reaches above 90% for the number of neighbors 4, 5 and 6. As the number of users increases, the system experiences intensive requirement, and the hit ratio steadily decreases. That is because, with the limited helper size, when there are more content requirements, the hit ratio decreases.

VII. CONCLUSIONS

In this paper, we investigate the content placement problem in BS-based wireless CDN system with constrained cooperation range. The evaluation confirms that the system performance is enhanced comparing to LFU without cooperation. The current work has some limitations. The cooperation range should be properly defined to balance load and avoid congestion. In the future, we aim to design mechanism which adaptively adjusts cooperation range based on dynamic system environments, and jointly optimize content placement with user request redirection.

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] D. Naylor, A. Finamore, I. Leontiadis, Y. Grunenberger, M. Mellia, M. M. Munafò, K. Papagiannaki, and P. Steenkiste, “The cost of the ”s” in HTTPS,” in *Proc. of ACM CoNEXT*, 2014, pp. 133–140.

[3] Z. Li and G. Simon, “In a telco-cdn, pushing content makes sense,” *IEEE Transactions on Network and Service Management*, vol. 10, no. 3, pp. 300–311, Sep. 2013.

[4] A. Wolman, G. M. Voelker, N. Sharma, N. Cardwell, A. R. Karlin, and H. M. Levy, “On the scale and performance of cooperative web proxy caching,” in *Proc. of ACM SOSP*, 1999, pp. 16–31.

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

[6] 3GPP, “Local ip access and selected ip traffic offload,” 3rd Generation Partnership Project (3GPP), Tech. Rep., Oct. 2014.

[7] X. Wang, M. Chen, T. Taleb, A. Ksentini, and V. Leung, “Cache in the air: exploiting content caching and delivery techniques for 5g systems,” *IEEE Communications Magazine*, vol. 52, no. 2, pp. 131–139, Feb. 2014.

[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*, Aug. 2015.

[9] D. Boscovic, F. Vakil, S. Dautovic, and M. Tosic, “Pervasive wireless cdn for greening video streaming to mobile devices,” in *2011 Proceedings of the 34th International Convention MIPRO*, May 2011, pp. 629–636.

[10] 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.

[11] 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.

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

[13] J. Leblet, Z. Li, G. Simon, and D. Yuan, “Optimal network locality in distributed virtualized data-centers,” *Computer communications*, vol. 34, no. 16, pp. 1968 – 1979, Oct. 2011.

[14] C. Andrieu, N. de Freitas, A. Doucet, and M. Jordan, “An introduction to mcmc for machine learning,” *Machine Learning*, vol. 50, no. 1-2, pp. 5–43, 2003.

[15] C. Joo, X. Lin, J. Ryu, and N. Shroff, “Distributed greedy approximation to maximum weighted independent set for scheduling with fading channels,” *IEEE/ACM Transactions on Networking*, vol. PP, no. 99, pp. 1–13, 2015.

[16] Z. Ming, M. Xu, and D. Wang, “Incan: In-network cache assisted enodeb caching mechanism in 4g LTE networks,” *Computer Networks*, vol. 75, pp. 367–380, Oct. 2014.