Distributed Wireless Video Caching Placement for Dynamic Adaptive Streaming

Chenglin Li
LTS4, EPFL, Switzerland
chenglin.li@epfl.ch

Pascal Frossard
LTS4, EPFL, Switzerland
pascal.frossard@epfl.ch

Hongkai Xiong
Shanghai Jiao Tong University, China
xionghongkai@sjtu.edu.cn

Junni Zou
Shanghai University, China
zoujn@shu.edu.cn

ABSTRACT

Caching at edge servers can smooth the temporal traffic variability and reduce the service load of base stations in wireless streaming. However, the assignment of the cached content for possibly multiple versions of different video sequences is still a challenging question in the context of adaptive streaming. In this paper, we propose a wireless video caching placement optimization problem for dynamic adaptive video streaming that properly takes into account the different rate-distortion (R-D) characteristics of the video sequences. Our objective is to minimize the expected video distortion by the optimal caching of compressed video sequences, such that the clients can effectively access video content storage and bandwidth constraints. We prove that our optimization problem is a submodular maximization problem subject to a knapsack constraint. A cost benefit greedy algorithm is developed to obtain an approximate solution with polynomial time complexity and theoretical approximation guarantees. Simulation results demonstrate significant video distortion reduction relative to different baseline caching placement schemes.

CCS Concepts

•Mathematics of computing → Submodular optimization and polymatroids; •Information systems → Multimedia streaming;

Keywords

Dynamic adaptive video streaming; video-on-demand; distributed caching; submodular function maximization

1. INTRODUCTION

With the extensive growth of global mobile data traffic and the widespread use of smart devices, wireless video streaming has experienced extensive growth and is commonly used for a wide range of applications, such as mobile video services. Meanwhile, the population of mobile users has become more heterogeneous in terms of mobile devices, requested contents, and network connectivity. Dynamic adaptive streaming over HTTP (DASH) is an effective method for video streaming over heterogeneous networks; it can improve the overall user satisfaction by offering several representations of the same video content to the different clients [1]. Each representation is encoded at a pre-defined bit rate and/or resolution such that users can be served by the most suitable representation in accordance with their requirements and heterogeneous network conditions.

At the same time, the network traffic presents a high temporal variability, which incurs congestion during peak traffic hours and under-utilization during off-peak hours. To reduce the peak traffic, caching (or pre-fetching) is proposed to utilize the storage space of edge servers across the network and to perform content placement during off-peak hours, thereby smoothing out the temporal traffic variability and reducing congestion and access latencies [2]. For wireless video streaming, caching at edge servers can greatly reduce the service load of the base station and replace the usually weak backhaul communications from the base station with high-speed local links to the edge servers [3].

In this paper, we formulate a distributed caching placement optimization problem for DASH based wireless video-on-demand (VoD) streaming system with proper consideration of the R-D properties of representations from different video sequences. Specifically, we target at maximizing the overall system utility in terms of the expected aggregate video utility incurred by distributed caching at edge servers under the edge servers' storage capacity constraints. This is achieved by the optimal assignment of DASH representations of multiple video sources to distributed edge servers. We further prove that the proposed distributed wireless video caching placement optimization problem is a submodular maximization problem subject to a knapsack constraint, which is NP-hard. Therefore, a cost benefit greedy algorithm is proposed in order to obtain an approximate solution with polynomial time complexity and theoretical approximation guarantees. Simulation results demonstrate significant video distortion reduction relative to common caching placement schemes, and reveal that the performance of the caching placement is greatly influenced by the R-D properties of different video contents.

NOSSDAV'16, May 13 2016, Klagenfurt, Austria

© 2016 ACM. ISBN 978-1-4503-4356-5/16/05. . . $15.00

DOI: http://dx.doi.org/10.1145/2910642.2910645

Other works in the literature address the video caching problem but without consideration of the video content information. For example, Jin *et al.* [4] apply caching to DASH streaming, and study the optimal transcoding and caching allocation scheme in media cloud in order to minimize the total operational cost of delivering on-demand adaptive video streaming, with the assumption that each mobile user accesses one edge server for video downloading. In [3], the wireless video caching problem with distributed edge servers is investigated to minimize the average downloading delay of users, where each video has only a single representation and each mobile user is connected to multiple distributed edge servers; through the cooperation among these edge servers the service load is largely transferred from the base station to edge servers. However, these works only focus on the operational-cost/rate perspective and thus neglect the video content information (e.g., the R-D properties) of the representations from different video contents. However, it is only by carefully considering the video content that the actual performance of the caching system can be properly evaluated.

The rest of this paper is organized as follows. Section 2 describes the system model and optimization formulation for the distributed wireless video caching placement problem. In Section 3, we show that it is a submodular maximization problem and develop an approximate caching algorithm. Section 4 presents experimental results, and evaluates the performance of the proposed algorithm in comparison to common caching strategies. The concluding remarks are given in Section 5.

2. PROBLEM FORMULATION

2.1 System Model

Consider a DASH based wireless VoD streaming system as illustrated in Fig. 1, suppose that F different video files each of which is encoded into M different representations are maintained at the base station. S edge servers with certain capabilities of pre-fetching and storing video content are deterministically placed in the wireless coverage region of the base station. These edge servers are geographically closer to the mobile users and thus can enable high-density spatial reuse of the wireless resources with high-speed localized communication, which is usually assumed to be much faster than the backhaul links connected to the base station [3]. For the VoD service with certain a priori knowledge of the video popularity distribution, some popular video files can be pre-fetched by the edge servers during the off-peak hours to relieve the service load of the base station and to replace the backhaul communication.

The distributed caching placement criterion is as follows. Whenever a mobile user make a playback request for a specific video, it attempts to download from one of its adjacent edge servers a representation with as higher quality as possible in accordance with the content placement and the available download link capacity. Generally, mobile users would like to subscribe a representation with higher quality, while for the same representation cached in multiple edge servers, they might want to download it from the edge server with highest transmission rate. That is, the user will first determine that whether there is a representation with highest bit rate available at one of its adjacent edge servers and the download of this representation can be supported by the link capacity. If yes, the user could download and playback that representation; otherwise, it would make a further determination for the representation with a smaller bit rate. This determination will continue until an available representation with certain bit rate is found at an edge server or the representation with the smallest bit rate is reached. Specifically, if the expected representation is cached at more than one edge servers, the mobile user will choose to download it from the edge server with the highest download link capacity such that the downloading delay is minimized. When no representation of the requested video is available at any adjacent edge server, the user has to turn to the base station for possible representation downloading. In this case, since the transmission rate of the backhaul links connected to the base station is much smaller than that of the local high-speed links provided by the edge servers, the user will only download the basic representation with the smallest encoding bit rate.

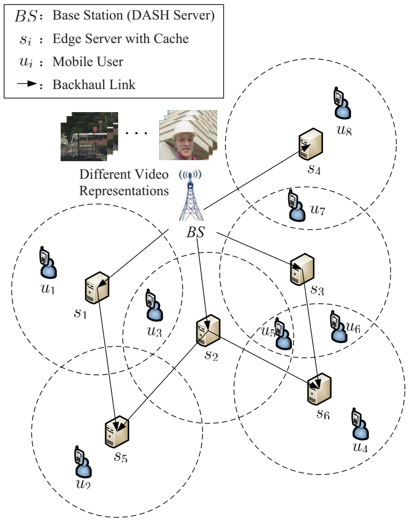

Figure 1: Example of the system layout, where mobile users are randomly distributed, while edge servers are connected to the base station with backhaul links and can be deterministically placed in the coverage region.

For video files, let $\mathcal{F} = \{1, 2, \ldots, F\}$ denote the set of F video files that are provided by the base station. For any video $f \in \mathcal{F}$, it will be encoded into $\mathcal{M} = \{1, 2, \ldots, M\}$ representations with each representation f_m's encoding bit rate being R_{f_m}. We further denote $f_{\mathcal{M}} = \{f_1, f_2, \ldots, f_M\}, \forall f \in \mathcal{F}$ as the set of M representations of video file f in a decreasing order of the encoding bit rate, i.e., $R_{f_i} > R_{f_j}, \forall 1 \leq i < j \leq M$. Therefore, the complete set including all the M representations for all the F video files can be denoted as $\mathcal{F}_{\mathcal{M}} = \cup_{f \in \mathcal{F}} f_{\mathcal{M}}$. Without loss of generality, in the following, we assume that each video file has the same time duration T. Such assumption is mainly proposed for the notational convenience, and could be easily relaxed by breaking a single longer video files into multiple files of the same length.

To illustrate the connection between edge servers and mobile users, the wireless network is defined by a bipartite graph $\mathcal{G}_{su} = (\mathcal{S}, \mathcal{U}, \mathcal{E}_{su})$, where $\mathcal{S} = \{1, 2, \ldots, S\}$ represents the set comprising S edge servers, $\mathcal{U} = \{1, 2, \ldots, U\}$ denotes the set of U mobile users, and $(s, u) \in \mathcal{E}_{su}$ indicates that a wireless communication link exists from edge server $s \in \mathcal{S}$ to mobile user $u \in \mathcal{U}$. The download link transmission rate of wireless link (s, u) is denoted by $c_{(s,u)}$. For each edge server $s \in \mathcal{S}$, the cache storage capability is constrained by the capacity B_s. A probability mass function $P_{u,f}$ is defined for video file $f \in \mathcal{F}$ and mobile user $u \in \mathcal{U}$, assuming that user u will make independent request to video file f with probability $P_{u,f}$.

Assuming that a representation of a video file can be either stored completely at the edge server, or not stored at all, the placement strategy can be represented by a bipartite graph $\mathcal{G}_{f_m,s} = (\mathcal{F}_{\mathcal{M}}, \mathcal{S}, \mathcal{E}_{f_m,s})$. Here, $\mathcal{F}_{\mathcal{M}}$ represents the complete set containing all representations for all the video files, and an edge $(f_m, s) \in \mathcal{E}_{f_m,s}$ denotes that f_m (i.e., the m-th representation of video file f) is stored in the cache of edge server s. To better understand the representation placement strategy as shown by the bipartite graph, we can further denote $\boldsymbol{A}_{FM\times S}$ as the $FM \times S$ adjacency matrix of $\mathcal{G}_{f_m,s}$, such that $\forall s \in \mathcal{S}$, $a_{f_m,s} = 1$ indicates that an edge $(f_m, s) \in \mathcal{E}_{f_m,s}$ exists and $a_{f_m,s} = 0$ otherwise. For a mobile user $u \in \mathcal{U}$, denote $\mathcal{S}(u)$ as its neighborhood of edge servers. $\mathcal{S}(u)$ is sorted in an decreasing order of the download link capacity, such that $(i)_u \in \mathcal{S}(u)$ represents the edge server with the i-th largest capacity of the link to mobile user u.

From the rate-distortion perspective, denote a general rate-distortion function $D_{\max} - D_f(R)$ as the distortion of video f with the encoding bit rate being R, where $D_{\max}$ and $D_f(R)$ represent a constant maximal distortion when no video is decoded and the distortion reduction (or quality improvement) after successful decoding the bitstream with bit rate R, respectively. Since distributed caching targets to shift the service load of the base station to edge servers, the performance measure is mainly determined by users' satisfaction on the service purely provided by cached contents, which can be derived as the expected average video distortion reduction experienced by mobile user u downloading from its adjacent edge servers:

$$\bar{D}_u = \sum_{f=1}^{F}\sum_{m=1}^{M}\sum_{i=1}^{|\mathcal{S}(u)|}\left[\prod_{n=1}^{m-1}\prod_{j=1}^{|\mathcal{S}(u)|}(1-a_{f_n,(j)_u})\right] \qquad (1)$$
$$\cdot\left[\prod_{j=1}^{i-1}(1-a_{f_m,(j)_u})\right]\cdot a_{f_m,(i)_u}\cdot P_{u,f}\cdot D_f(R_{f_m})$$

In Eq. (1), the term $[\prod_{n=1}^{m-1}\prod_{j=1}^{|\mathcal{S}(u)|}(1-a_{f_n,(j)_u})][\prod_{j=1}^{i-1}(1-a_{f_m,(j)_u})]\cdot a_{f_m,(i)_u} = 1$ is the indicator function defined over the set of feasible placement matrix $\boldsymbol{A}_{FM\times S}$ for the case that the m-th representation of video file f is the best representation that user u could find in its neighboring edge servers (specifically, $[\prod_{n=1}^{m-1}\prod_{j=1}^{|\mathcal{S}(u)|}(1-a_{f_n,(j)_u})] = 1$ indicates that no representation with index smaller than m is at any of the adjacent edge servers), and this representation is at the cache of edge server $(i)_u$ (specifically, $[\prod_{j=1}^{i-1}(1-a_{f_m,(j)_u})] = 1$ indicates that this representation is not available at any of the edge servers with larger download link rate than edge server $(i)_u$).

2.2 Optimization Problem Formulation

The distributed wireless video caching placement problem for DASH streaming can be summarized as: for a given representation set of source video files, file popularity distribution, edge server storage capacity and the wireless network topology, how to place the representations of all the video files in the distributed edge servers such that the total system utility is maximized subject to the caching capacity constraint of each edge server. If each video file has only one representation (i.e., DASH is not applied) and each mobile user has only access to one edge server, the optimal placement strategy becomes simple and straightforward. That is, each edge server should cache as many most popular video files as possible until its storage is full. However, for the case of dense edge server deployment within which each mobile user can have access to more than one edge server, the optimal content placement strategy becomes highly nontrivial. When DASH streaming is taken into account and applied to the VoD system, one video file is further encoded and stored as multiple representations with different bit rates. The optimal placement problem in this case becomes even more complicated.

Mathematically, this problem can be formulated as:

$$\max_{\boldsymbol{A}_{FM\times S}\in\{0,1\}^{FM\times S}} \quad \sum_{u=1}^{U}\bar{D}_u \qquad (2a)$$
$$\text{s.t.} \quad \sum_{f=1}^{F}\sum_{m=1}^{M} a_{f_m,s}\cdot R_{f_m}\cdot T \le B_s, \forall s\in\mathcal{S} \qquad (2b)$$

The objective in Eq. (2a) is to maximize the aggregate video distortion reduction of all mobile users (equivalent to minimizing the average video distortion per user, i.e., min $[D_{\max} - \sum_{u=1}^{U}\bar{D}_u/U]$), and the decision variable is the representation placement strategy represented by the adjacency matrix $\boldsymbol{A}_{FM\times S} \in \{0,1\}^{FM\times S}$. The constraint in Eq. (2b) is the cache capacity requirement constrained by the storage of edge servers, where B_s is the storage capacity of edge sever s and T is the time duration of video sequences.

In general, the proposed optimization problem in Eq. (2) can be solved through integer programming. However, in the next section, we will demonstrate that it is a constrained maximization problem for a submodular function[1], which is NP-hard in general cases and needs very high computational complexity to achieve the optimal solution.

3. SUBMODULARITY AND APPROXIMATION ALGORITHM

3.1 Submodular Maximization

In accordance with the DASH based wireless video caching placement problem, the finite ground set can be viewed as:

$$\mathcal{V} = \{\mathcal{V}_1, \ldots, \mathcal{V}_s, \ldots \mathcal{V}_S\}, \qquad (3)$$
$$\mathcal{V}_s = \{v_{1,1}^s, \ldots, v_{1,M}^s, \ldots, v_{f,m}^s, \ldots, v_{F,1}^s, \ldots, v_{F,M}^s\},\ \forall s\in\mathcal{S}$$

where the ground set is partitioned into S disjoint subsets. Each subset $\mathcal{V}_s$ denotes the full set of all representations of all files that may be cached on the edge server s, and the element $v_{f,m}^s$ represents the placement of the m-th representation of video file f on the cache of the edge server s. For a given adjacency matrix $\boldsymbol{A}_{FM\times S}$, the corresponding representation placement set $\mathcal{A} \subseteq \mathcal{V}$ can be defined in such a way that $v_{f,m}^s \in \mathcal{A}$ corresponds to the case $a_{f_m,s} = 1$ and

[1] Let $\mathcal{V}$ be a finite ground set, and a set function $g : 2^{\mathcal{V}} \to \mathbb{R}$ is submodular iff $g(\mathcal{X}\cup\{v\}) - g(\mathcal{X}) \ge g(\mathcal{Y}\cup\{v\}) - g(\mathcal{Y})$ for any sets $\mathcal{X} \subseteq \mathcal{Y} \subseteq \mathcal{V}$ and for any element $v \in (\mathcal{Y}\setminus\mathcal{X})$.

vice versa. In terms of set function, the expected distortion reduction function in Eq. (1) can be rewritten as:

$$\bar{D}_u(\mathcal{A}) = \sum_{f=1}^{F}\sum_{m=1}^{M}\sum_{i=1}^{|\mathcal{S}(u)|}\left[\prod_{n=1}^{m-1}\prod_{j=1}^{|\mathcal{S}(u)|}(1-\mathbf{1}|_{v_{f,n}^{(j)_u}\in\mathcal{A}})\right] \tag{4}$$

$$\cdot\left[\prod_{j=1}^{i-1}(1-\mathbf{1}|_{v_{f,m}^{(j)_u}\in\mathcal{A}})\right]\cdot\mathbf{1}|_{v_{f,m}^{(i)_u}\in\mathcal{A}}\cdot P_{u,f}\cdot D_f(R_{f_m})$$

where $\mathbf{1}|_{v\in\mathcal{A}}$ is an indicator function, the value of which is 1 if $v \in \mathcal{A}$ and 0 otherwise.

Proposition 1. *The objective function in Eq. (2a) is a monotone submodular function over the ground set $\mathcal{V}$ defined in Eq. (3).*

Proof. Denote the equivalent set function of Eq. (2a) as $D(\mathcal{A}) = \sum_{u=1}^{U}\bar{D}_u(\mathcal{A})$. It is easy to observe that $D(\mathcal{A})$ is monotone non-decreasing.

According to the property of submodularity, the summation over a set of submodular functions is also submodular. Thus, to prove the submodularity of $D(\mathcal{A})$, it is only required to prove that for every mobile user u the set function $\bar{D}_u(\mathcal{A})$ is submodular.

Consider any two placement sets $\mathcal{A} \subseteq \mathcal{A}' \subseteq \mathcal{V}$. For some edge server $s = (i)_u, 1 \leq i \leq |\mathcal{S}(u)|$, suppose adding a new element $v_{f,m}^{s} = v_{f,m}^{(i)_u} \in \mathcal{V} \setminus \mathcal{A}'$ to both placement sets. That is, adding the m-th representation of video file f into the cache of edge server $s = (i)_u$. Next, we consider the following two cases.

i) There exists $v_{f,n'}^{(j')_u} \in \mathcal{A}'$ with $n' \leq m$, i.e., according to the placement set $\mathcal{A}'$ mobile user u downloads a better or equal quality representation n' of video file f from the edge server $(j')_u$. In this case, it can be derived from Eq. (4) that $\bar{D}_u(\mathcal{A}' \cup \{v_{f,m}^{(i)_u}\}) - \bar{D}_u(\mathcal{A}') = 0$. On the other hand, due to the monotonicity, for the placement set $\mathcal{A}$ we always have $\bar{D}_u(\mathcal{A} \cup \{v_{f,m}^{(i)_u}\}) - \bar{D}_u(\mathcal{A}) \geq 0$. Therefore, the relationship of both marginal values is given by $\bar{D}_u(\mathcal{A}' \cup \{v_{f,m}^{(i)_u}\}) - \bar{D}_u(\mathcal{A}') \leq \bar{D}_u(\mathcal{A} \cup \{v_{f,m}^{(i)_u}\}) - \bar{D}_u(\mathcal{A})$.

ii) There exists $v_{f,n'}^{(j')_u} \in \mathcal{A}'$ with $n' > m$, i.e., according to the placement set $\mathcal{A}'$ mobile user u downloads a worse quality representation n' of video file f from the edge server $(j')_u$. In this case, it can be derived from Eq. (4) that $\bar{D}_u(\mathcal{A}' \cup \{v_{f,m}^{(i)_u}\}) - \bar{D}_u(\mathcal{A}') = P_{u,f}[D_f(R_{f_m}) - D_f(R_{f'_n})]$. On the other hand, for the placement set $\mathcal{A}$, since $\mathcal{A} \subseteq \mathcal{A}'$, mobile user u can only downloads representation n of file f from the edge server $(j)_u$ with $n \geq n'$. Thus, the resulting marginal value is $\bar{D}_u(\mathcal{A} \cup \{v_{f,m}^{(i)_u}\}) - \bar{D}_u(\mathcal{A}) = P_{u,f}[D_f(R_{f_m}) - D_f(R_{f_n})]$. Since $n \geq n'$, we have $R_{f'_n} \geq R_{f_n}$ and thus $D_f(R_{f_{n'}}) \geq D_f(R_{f_n})$. Therefore, the relationship of both marginal values is given by $\bar{D}_u(\mathcal{A}' \cup \{v_{f,m}^{(i)_u}\}) - \bar{D}_u(\mathcal{A}') \leq \bar{D}_u(\mathcal{A} \cup \{v_{f,m}^{(i)_u}\}) - \bar{D}_u(\mathcal{A})$.

For both cases, the marginal value decreases as the set becomes larger, which satisfies the submodularity definition. Hence, the submodularity is proved. □

In Proposition 1, we have justified that Eq. (2a) is a submodular function. Further observing the cache storage constraint of edge server $s \in \mathcal{S}$ in Eq. (2b), each element $v_{f,m}^{s} \in \mathcal{A}$ (corresponding to the case $a_{f_m,s} = 1$ in $\boldsymbol{A}_{FM\times S}$) has a non-uniform cost of $R_{f_m} \cdot T$, and s has a storage budget of B_s. Such constraint can be viewed as a knapsack constraint on the subset $\mathcal{V}_s \in \mathcal{S}$, and thus the set of cache storage constraints of all the edge servers forms a knapsack constraint on the finite ground set $\mathcal{S}$. Therefore, the distributed caching placement problem in Eq. (2) is a submodular maximization problem subject to a knapsack constraint, which is generally NP-hard and requires very high computational complexity to achieve the optimal solution by either integer programming or other methods [5].

Algorithm 1 k-Cost benefit (k-CB) greedy algorithm

```
For all initial sets $\mathcal{A}^0 \subseteq \mathcal{V}$ such that $|\mathcal{A}^0| = k$, implement the following cost benefit greedy procedure:
  Initialization:
    1) Set $\mathcal{V}^0 = \mathcal{V}$ and $t = 1$.
  Greedy Search Iteration: (at step $t = 1, 2, 3, \dots$)
    1) Given a partial solution $\mathcal{A}^{t-1}$, find
       $\theta_t = \max_{v_{f,m}^{s}\in\mathcal{V}^{t-1}\setminus\mathcal{A}^{t-1}} \frac{D(\mathcal{A}^{t-1}\cup\{v_{f,m}^{s}\}) - D(\mathcal{A}^{t-1})}{R_{f_m}\cdot T}$   (5)
       with
       $v_{f_t,m_t}^{s_t} = \arg\max_{v_{f,m}^{s}\in\mathcal{V}^{t-1}\setminus\mathcal{A}^{t-1}} \frac{D(\mathcal{A}^{t-1}\cup\{v_{f,m}^{s}\}) - D(\mathcal{A}^{t-1})}{R_{f_m}\cdot T}$   (6)
  Update and Determination:
    1) Set $\mathcal{A}^t = \mathcal{A}^{t-1} \cup \{v_{f_t,m_t}^{s_t}\}$, and $\mathcal{V}^t = \mathcal{V}^{t-1}$, if
       $\sum_{f=1}^{F}\sum_{m=1}^{M} \mathbf{1}|_{v_{f,m}^{s_t}\in(\mathcal{A}^{t-1}\cap\mathcal{V}_{s_t})\cup\{v_{f_t,m_t}^{s_t}\}} \cdot R_{f_m} \cdot T \leq B_{s_t};$   (7)
       otherwise, set $\mathcal{A}^t = \mathcal{A}^{t-1}$, and $\mathcal{V}^t = \mathcal{V}^{t-1} \setminus \{v_{f_t,m_t}^{s_t}\}$.
    2) If $\mathcal{V}^t \setminus \mathcal{A}^t \neq \emptyset$, set $t = t + 1$ and return to the greedy search iteration; otherwise, stop the iteration.
The solution is obtained and output as $\mathcal{A}$, which has the largest value of the objective function $D(\mathcal{A}) = \sum_{u=1}^{U}\bar{D}_u(\mathcal{A})$ over all the possible choices of the initial sets $\mathcal{A}^0 \subseteq \mathcal{V}$.
```

3.2 Approximate Algorithm

To efficiently solve the submodular maximization problem with polynomial time complexity and theoretical approximation guarantees, the k-cost benefit (k-CB) greedy algorithm is developed as shown in Algorithm 1, with k indicating the size of the initial set. Specifically, the proposed k-CB greedy algorithm considers all feasible initial sets $\mathcal{A}_0 \subseteq \mathcal{V}$ of cardinality k. Starting from any initial set $\mathcal{A}_0$, at step t, the cost benefit greedy procedure iteratively searches over the remaining set $\mathcal{V}^{t-1} \setminus \mathcal{A}^{t-1}$ and inserts into the partial solution $\mathcal{A}^{t-1}$ an element according to Eqs. (6) and (7), until the remaining set reduces to an empty set. In other words, the cost benefit procedure adds at each iteration an element that maximizes the marginal benefit $D(\mathcal{A}^{t-1} \cup \{v_{f,m}^{s}\}) - D(\mathcal{A}^{t-1})$ and cost $R_{f_m} \cdot T$ ratio among all elements still affordable with the remaining storage budget until no more element can be added. The proposed k-CB greedy algorithm then enumerates all initial sets $\mathcal{A}_0 \subseteq \mathcal{V}$ of cardinality k, augments each of them following the cost benefit greedy procedure, and selects the initial set achieving the largest value of the objective function $D(\mathcal{A}) = \sum_{u=1}^{U}\bar{D}_u(\mathcal{A})$ and sets its solution set as the final placement set $\mathcal{A}$. For the special case of $k = 0$, the algorithm reduces to a simple cost benefit greedy algorithm starting with $\mathcal{A}_0 = \emptyset$.

In terms of computational complexity, the running time of the proposed k-CB greedy algorithm is $O((SFM)^{k+2}U)$, which indicates a polynomial time complexity. As the value of k increases, the running time of the proposed algorithm becomes larger while the performance improves. As shown in [6], when $k \geq 3$, the theoretical worst-case performance guarantee of the proposed algorithm is $1 - 1/e$, i.e., its solution achieves at least the ratio $1 - 1/e$ of the optimal objective value.

4. EXPERIMENTAL EVALUATION

In this section, we evaluate the performance of the proposed k-CB greedy algorithm under different video file and network settings, and illustrate its effectiveness over the other two other schemes: 1) *Femto-Greedy*, the greedy algorithm proposed in [3] to minimize the average downloading delay of users for wireless video content delivery through distributed caches; and 2) *Popular-Cache*, each edge server pre-fetches the most popular video files allowed by its storage capacity. Note that one of our main contributions is the optimal assignment of the cached contents to distributed edge servers for multiple DASH representations, while the femto-greedy and popular-cache algorithms only consider one single representation for each video. To have a "fair" comparison with the proposed algorithm, here, the femto-greedy and popular-cache algorithms are simply extended to facilitate multiple DASH representations as follows. The algorithm is first individually applied to the representations of all videos with the same representation index m, i.e., $\cup_{f \in \mathcal{F}} f_m$. Then, we enumerate all the possible representation index $m = 1, 2, \ldots, M$ and choose the one achieving the largest aggregate distortion reduction as the final solution.

4.1 Settings

First, we consider a wireless network with $S = 3$ edge servers uniformly placed and $U = 20$ users randomly distributed in a 100 m × 100 m square region. Assume that the connectivity range (effective transmission range) of each edge server is 50 m, and the network connectivity graph is accordingly shown in Fig. 2(a). Three test video sequences ($F = 3$, *Crowd Run*, *Tractor*, and *Sunflower*) with 1080p resolution (1920×1080), available at [7], are selected as the video files needed for caching. These three test video sequences correspond to different content types, i.e., dense object motion for *Crowd Run* sequence, camera movement and medium object motion for *Tractor* sequence, and small object motion for *Sunflower* sequence, respectively. The distortion versus encoding bit rate curves of these three sequences are illustrated in Fig. 2(b). Suppose that the time duration of each video clip is $T = 6$ s, and the constant maximal distortion is set as $D_{\max} = 500$. At frame rate of 30 fps, we further encode each video sequence into $M = 3$ representations with encoding rate being $\{3R,\ 2R,\ R\}$ and $R = 1000$ Kbps. Accordingly, the distortion reduction in MSE after successful decoding all the representations is listed in Table 1. Generally, it can be seen that for video with small object motion (e.g., *Sunflower*), the representation with smallest encoding bit rate introduces a large distortion reduction (e.g., 483.2 with encoding bit rate of 1000 Kbps) while increasing the encoding bit rate cannot incur significant additional distortion reduction (e.g., only 11.4 additional distortion reduction between encoding bit rate of 1000 Kbps and 3000 Kbps); and vice versa. The storage capacity for each edge server is set to $B_s = 4RT$. For the video file popularity, we further assume that these three sequences follow a Zipf distribution with parameter 0.56 [8], i.e., the requesting probabilities of *Crowd Run*, *Tractor*, and *Sunflower* sequences are 0.45, 0.31, and 0.24, respectively.

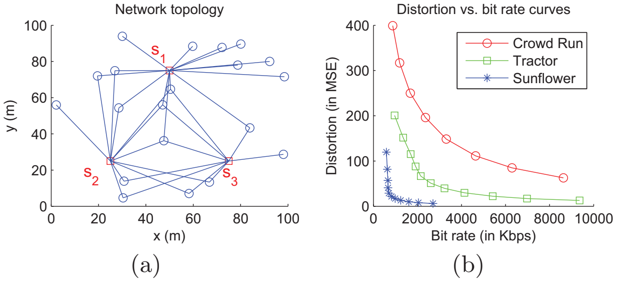

Figure 2: (a) Network connectivity graph with 3 edge servers and 20 users. (b) Distortion vs. encoding bit rate curves of the three sequences.

Table 1: Distortion reduction (in MSE) after decoding representations of different video sequences

Bit rate	3000 Kbps	2000 Kbps	1000 Kbps
Crowd Run	335.9	275.4	133.3
Tractor	456.3	419.8	303.7
Sunflower	494.6	491.9	483.2

4.2 Results

In Table 2, we compare the actual running time in simulation, theoretical computational complexity, and the average distortion reduction per user achieved by different algorithms. It can be seen from Table 2 that in terms of average distortion reduction per user, the proposed k-CB greedy algorithm generally outperforms the other two comparison algorithms. Specifically, popular-cache and femto-greedy algorithms result in 0.75 and 0.86 approximation solution of the optimal solution that is obtained by the exhaustive search over all possible feasible solution sets. For the case of $k = 0$ and $k = 1$, the proposed algorithm advances the approximation ratio to 0.96 and 0.99, respectively. When k becomes large, e.g. $k = 2,\ 3$, the proposed algorithm can even achieve the optimal solution. From the perspective of running time, the complexity of exhaustive search is exponential and becomes not feasible with the increment of the number of video files and the network scale. In contrast, the proposed k-CB greedy algorithm has polynomial time complexity and the running time will be reduced as k decreases. Specifically, when $k = 0$, the proposed algorithm achieves quadratic time complexity the same as femto-greedy algorithm but the performance is better. Although the running time of poplar-cache algorithm is smallest, its performance is also the worst. Therefore, we can seek the tradeoff between the algorithm performance and the running time by adapting the value of k of the proposed k-CB greedy algorithm. In practice, to have a near optimal approximation solution with affordable running time, we could set k to 0 or 1 for large scale network.

To gain further insight into the difference between different algorithms, in Table 3, we list the placement sets of edge servers obtained by the optimal solution (3- or 2-CB greedy), femto-greed and popular-cache algorithms. Within each vector, the three rows from top to bottom correspond

Table 2: Comparison on computational complexity and performance of different algorithms

Algorithm	Time (s)	Computation complexity	$\frac{1}{U}\sum_{u=1}^{U}\bar{D}_u$
Exhaust search	2068.9	Exponential	361.8 (100%)
3-CB Greedy	301.2	$O((SFM)^5U)$	361.8 (100%)
2-CB Greedy	38.2	$O((SFM)^4U)$	361.8 (100%)
1-CB Greedy	2.6	$O((SFM)^3U)$	357.4 (99%)
0-CB Greedy	0.3	$O((SFM)^2U)$	347.5 (96%)
Femto-Greedy	0.3	$O((SFM)^2U)$	312.0 (86%)
Popular-Cache	0.05	$O(SFM)$	270.6 (75%)

Table 3: Placement strategy for edge servers $S_1 - S_3$ obtained by different algorithms

Algo.	S_1	S_2	S_3
Opt.	$\begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}$
Femto.	$\begin{pmatrix} 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}$	$\begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}$	$\begin{pmatrix} 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}$
Pop.	$\begin{pmatrix} 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}$	$\begin{pmatrix} 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}$	$\begin{pmatrix} 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}$

to *Crowd Run*, *Tractor*, and *Sunflower* sequences, while the three columns from left to right stand for the encoding rate of $\{3R,\ 2R,\ R\}$. The fundamental reason why the proposed algorithm outperforms the others can be revealed as follows. In addition to the consideration of video file popularity and the cooperation among different edge servers, the caching decision of representations for different videos can be further adapted by the proposed algorithm according to the video content information. For video sequence with small motion (e.g., *Sunflower*), the proposed algorithm only allocates the basic representation with smallest bit rate R at each edge server, while for video sequences with larger motion (e.g., *Crowd Run* and *Tractor*), representations with larger bit rate $2R$ or $3R$ are allocated at some edge servers to gain larger distortion reduction.

Next, we conduct simulations for larger scale settings, with $F = 10$ video files and $M = 3$ representations for each video, $U = 300$ users randomly placed in a 400 m × 400 m square region. The connectivity range and the storage capacity of each edge server are set to 70 m and $B_s = 4RT$, and all the other parameters are the same as previous. In Fig. 3, the average distortion reduction per user under different algorithms is shown for different edge server numbers $S = 16,\ 20,\ 25$, respectively. Again, it is verified that the proposed algorithm outperforms the other two comparison algorithms for different number of S. It can also be observed that as the number of S increases, the performance of popular-cache would not change since each edge server always separately pre-fetches the most popular video files, while femto-greedy and the proposed algorithm could gradually benefit from a denser deployment of the edge servers within the wireless region.

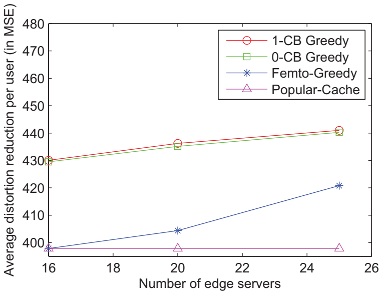

Figure 3: Average distortion reduction per user (in MSE) vs. the number of edge servers.

5. CONCLUSIONS

In this paper, we studied a distributed caching placement optimization problem for DASH based wireless VoD streaming system in order to maximize the expected aggregate video distortion reduction. The problem proved to be a submodular maximization problem subject to a knapsack constraint. An approximate caching algorithm has been provided, as a tradeoff between approximation performance and running time. Experimental results have shown significant video distortion reduction relative to existing schemes. We also found that the performance of the caching placement is greatly affected by the R-D properties of different video contents. Future research directions include the distributed caching placement strategy adapted to other QoE metrics, such as startup delay and quality variations.

6. ACKNOWLEDGMENTS

This work was supported in part by the NSFC, under grants 61425011, U1201255, 61271218, 61529101, 61501293, 61472234, the "Shu Guang" project (13SG13), and the China Postdoctoral Science Foundation.

7. REFERENCES

[1] D. H. Lee, C. Dovrolis, and A. C. Begen. Caching in HTTP adaptive streaming: Friend or foe? In *Proc. ACM NOSSDAV*, 2014.

[2] M. A. Maddah-Ali and U. Niesen. Fundamental limits of caching. *IEEE Transactions on Information Theory*, 60(5):2856–2867, May 2014.

[3] 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*, 59(12):8402–8413, Dec. 2013.

[4] Y. Jin, Y. Wen, and C. Westphal. Optimal transcoding and caching for adaptive streaming in media cloud: An analytical approach. *IEEE Transactions on Circuits and Systems for Video Technology*, 25(12):1914–1925, Dec. 2015.

[5] A. Krause and D. Golovin. Submodular function maximization. *Tractability: Practical Approaches to Hard Problems*, 3:19, 2012.

[6] M. Sviridenko. A note on maximizing a submodular set function subject to a knapsack constraint. *Operations Research Letters*, 32(1):41–43, 2004.

[7] Xiph.org video test media. [online]. Available: http://media.xiph.org/video/derf/.

[8] M. Zink, K. Suh, Y. Gu, and J. Kurose. Characteristics of youtube network traffic at a campus network–measurements, models, and implications. *Computer networks*, 53(4):501–514, 2009.