Mobility-Aware Caching Specific to Video Services in Hyper-Dense Heterogeneous Networks

Zhenya Liu(✉), Xi Li, Hong Ji, and Heli Zhang

Key Laboratory of Universal Wireless Communications, Ministry of Education,
Beijing University of Posts and Telecommunications,
Beijing, People's Republic of China
{liuzhenya,lixi,jihong,zhangheli}@bupt.edu.cn

Abstract. Caching at the network edge has emerged as a promising technique to cope with the dramatic increase of mobile data traffic. It is noted that different types of video applications on mobile devices have different requirements for cached contents, thus corresponding caching policies should be developed accordingly. In hyper-dense heterogeneous networks, due to the user mobility and limited connection duration, the user often could not download the complete cached contents from an associated SBS before it moves away, which makes the design of caching strategy more challenging. In this paper, we propose two different caching strategies to adapt to multimedia applications of different video contents. For ordinary network video files, coded caching is used to increase the efficiency of content access. The caching problem is formulated as an optimization problem to minimize the average transmission cost of cached contents. We first present an optimal caching strategy based on the critical value of validity period of user requests. Then, for the validity period greater than its critical value, an iterative optimization on the basis of the above optimal solution is performed. For typical streaming video, uncoded video fragments is considered to be stored in the caches to meet the needs of online viewing. The principle of the proposed caching scheme is to cache data chunks in advance according to the sequences of SBSs passed by the user based on the mobility prediction results. Simulation results indicate that the proposed mobility-based caching performs better than the existing popularity-based caching scheme.

Keywords: Mobility-aware caching · Video services · Transmission cost · Heterogeneous networks (HetNets)

1 Introduction

With the popularization of smart terminals and the diversification of multimedia applications, mobile data traffic has exhibited unprecedented growth worldwide. This traffic growth in mobile networks will result in higher transmission

X. Liu et al. (Eds.): ChinaCom 2018, LNICST 262, pp. 480–490, 2019.
https://doi.org/10.1007/978-3-030-06161-6_47

delay and energy consumption. To cope with the stern challenge, researchers have proposed to deploy SBSs together with macro-cell base stations (MBSs) in existing networks to boost the network capacity. However, the overloaded and costly backhaul links connecting the SBSs with the core network becomes the bottleneck in improving network performance.

Caching popular contents at the SBSs equipped with storage facility has been proposed to relieve congestion in the backhaul links [1]. By storing the frequently requested contents in SBSs cache in advance, this technique not only avoids redundant file retrieval over the backhaul links, but reduces the user experienced delay. Motivated by this, content caching have been widely studied in small cell networks. However, many works do not consider user mobility [2,3]. In realistic environments, the users are not stationary and their association with the SBSs may change during data transfer, which makes the design of caching strategy more intractable.

In hyper-dense HetNets, the association between the user and the SBSs will change more frequently. Reference [4] has shown that analyzing and exploiting the mobility patterns of users, the mobility-aware caching can markedly improve the efficiency of caching. In [5], the authors formulate the content caching in small-cell networks as an optimization problem, with the goal of maximize the caching utility. In [6], the authors first model the user sojourn time as a random variable that obeys the exponential distribution, and then propose a file allocation strategy based on coded caching. In [7], the authors research the impact of user mobility on content caching aimed at minimizing the load of the macrocells. In [8], the authors develop a novel algorithm for content placement at the cache based on estimated popularity.

Although these works have taken user mobility into account, most of them assume that all data of requested file stored in the cache of connected SBS can be downloaded, once the user established a connection with it. In practice, the user may only get parts of the cached contents from the associated SBS during each connection due to the limit of connection duration. Moreover, in recent years, various multimedia applications are emerging endlessly to meet the different user needs. We notice that different types of video applications on user terminals have different requirements for cached contents. Therefore, corresponding caching policies should also be developed.

In this paper, we propose two different caching strategies specific to different video services. For ordinary network video files, coded caching is introduced to increase the efficiency of content access. The caching problem is formulated as an optimization problem to minimize the average transmission cost of cached contents. We first present an optimal caching strategy at the critical value of validity period of user requests. Then, for the validity period greater than its critical value, an iterative optimization based on the aforementioned optimal result is implemented to maximize the average amount of coded data delivered by local caches. For typical streaming video, storing uncoded video segments at SBSs is considered to meet the needs of online viewing. The idea of the proposed caching scheme is to cache data chunks in advance according to the sequences of SBSs traversed by the user. Simulation results show that our proposed mobility-based caching performs better than maximum popularity caching.

2 System Model

2.1 Network Model

Consider a heterogeneous network for video delivery like the one depicted in Fig. 1, which consists of one MBS and a set $\mathcal{N}$ of N SBSs deployed in a macro-cell. Each SBS$_n$, $n \in \mathcal{N}$ is equipped with a cache of storage size C_n (bytes). The coverage areas of the SBSs may overlap each other, and a user may be concurrently covered by multiple SBSs. Since the SBS coverage area is relatively small, the user may repeatedly move in and out of the small cells and thus connect to different SBSs at different times.

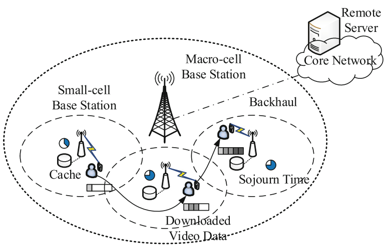

Fig. 1. Graphical illustration of heterogeneous network and user trajectory.

User preference to content files can be learned by analyzing previous statistics of user requests, and it is assumed to be known and fixed within a time period. We consider that each user independently requests a item from the content library $\mathcal{F}$ consisting of F video files, and file popularity follows Zipf distribution [9].

2.2 Video Service Model

Given the different requirements of multimedia applications, the cache space of each SBS is divided into two areas to store different contents, thus providing mobile users with different video services. For ordinary network video files, storing the encoded data of video files is considered to increase the efficiency of content access. By appropriate coding, the requested file can be successfully recovered when the total amount of downloaded coded data in any order is at least the size of the original file [10]. For typical streaming video, in-order packet delivery should be guaranteed. Since the storage buffers at the SBSs specific to such applications is limited, only some video fragments are expected to be placed in the cache to improve cache hit probability. We consider splitting each item into some data chunks with the same size, each of which is identified by a sequence number. The user collects data chunks sequentially from the encountered SBSs to meet the needs of viewing streaming video online.

2.3 Mobility Model

The user mobility is modeled from two dimensions of time and space. We first divide time into identical time intervals, and each time interval corresponds to the shortest duration that the SBS is accessed. Then, several important locations that are frequently visited are identified in the macro-cell, e.g., crowded crossroad, shopping center, stadium, etc. These important locations can be extracted by using clustering algorithms from previous trajectories of the user, and each location may be covered by multiple SBSs. Besides, there is also a non-important location covered by all remaining SBSs.

In the coded caching, we consider delayed offloading scheme [11]. To meet QoS requirements, we associate each user request with a period of validity. That is, each request must be completely served within T time intervals by the encountered SBSs once it is initiated. We refer to the sequence of visited locations during the validity period as movement pattern of the user, i.e. $r_w = \{v_1, v_2, \cdots, v_T\}$, where v_i represents the location visited at i-th interval. We denote the set of all possible movement patterns with $\mathcal{W}$. The probability that a user takes r_w, $w \in \mathcal{W}$ can be derived as follows:

$$q_w = p(v_1) \prod_{i=1}^{T-1} p(v_{i+1}|v_i) \tag{1}$$

where $p(v_1)$ denotes the probability that the user appears in location v_1 when initiating a video request, and $p(v_{i+1}|v_i)$ denotes the transition probability between v_i and v_{i+1}. These probabilities can be estimated by leveraging previous time statistics.

In the chunk-based uncoded caching, our work focuses on utilizing either the specified or the predicted mobility information to facilitate effective content placement at SBSs. We refer to the sequence of SBSs accessed within the coverage of the identified locations as movement trajectory of the user. We assume that the prediction of user trajectory is performed by the mobility prediction entity deployed at the MBS. In the following, two different kinds of caching strategies are presented.

3 Proposed Coded Caching Strategy

3.1 Problem Formulation

We use $\mathcal{X} = \{x_{n,i}|n \in \mathcal{N}, i \in \mathcal{F}\}$ to denote the caching strategy, where $x_{n,i}$ indicates the amount of coded data of video file f_i stored in the cache of SBS_n. Considering that there are differences between SBSs in the deployed bandwidth and the average workload, B_n is used to represent the average amount of data that SBS_n can deliver to a user within a time interval. In a movement pattern r_w, a user may be connected to the same SBS multiple times. Let $\mathcal{N}_w$ represent the subset of SBSs that are encountered in r_w, and $a_{w,n}$ is introduced to denote the number of time intervals that $\text{SBS}_n \in \mathcal{N}_w$ is accessed in r_w. During the j-th connection with SBS_n, the non-redundant amount of coded data of file f_i which can be downloaded within this interval is given by

$$d_{n,i}^{(j)} = \max\{\min\{x_{n,i} - (j-1)B_n, B_n\}, 0\} \tag{2}$$

For a user requesting file f_i and taking movement pattern r_w, the total amount of coded data downloaded from the local encountered SBSs can be expressed as follows:

$$u_{i,w} = \min\{\sum_{n\in\mathcal{N}_w}\sum_{j=1}^{a_{w,n}} d_{n,i}^{(j)}, s_i\} = \min\{\sum_{n\in\mathcal{N}} \min\{x_{n,i}, B_n a_{w,n}\}, s_i\} \tag{3}$$

where s_i is the size of file f_i. If $u_{i,w}$ is less than s_i, the remaining video segments need to be downloaded from the remote server. Then, the amount of coded data downloaded from the core network over backhaul link for file f_i is equal to $s_i - u_{i,w}$.

Obviously, serving a user request from the SBSs cache and the remote server will incur different levels of transmission costs to the operator, such as the energy consumed at the SBSs or the traffic generated in the backhaul network. Suppose that the cost of transmitting the unit of coded data volume from the cache of the encountered SBSs is ω_0, and the cost from the core network over backhaul link is ω_1 ($\omega_1 > \omega_0$). Then, in the case that the user takes the movement pattern r_w, the download cost of requested video file f_i is given by $(u_{i,w}\omega_0 + (s_i - u_{i,w})\omega_1)$.

Our goal is to find the optimal caching strategy to minimize the average transmission cost of the requested items. The optimization problem can be formulated as follows:

$$\begin{aligned}\min_{\mathcal{X}} \Omega(\mathcal{X}) &= \sum_{w\in\mathcal{W}} q_w \sum_{i\in\mathcal{F}} p_i \cdot (s_i\omega_1 - u_{i,w}(\omega_1 - \omega_0)) \\ &= \omega_1 \sum_{i\in\mathcal{F}} p_i s_i - (\omega_1 - \omega_0)\sum_{w\in\mathcal{W}}\sum_{i\in\mathcal{F}} q_w p_i u_{i,w}\end{aligned} \tag{4}$$

$$s.t.\ \ x_{n,i} \in [0, s_i],\ \forall n\in\mathcal{N},\ \forall i\in\mathcal{F};\quad \sum_{i\in\mathcal{F}} x_{n,i} \le C_n^{'},\ \forall n\in\mathcal{N}. \tag{5}$$

where $C_n^{'}$ denotes the capacity used to store MDS-encoded data of ordinary video files in the cache of SBS_n, satisfying $C_n^{'} < C_n$. Let $\Phi(\mathcal{X})$ denote the average amount of coded data delivered by the encountered SBSs, which is derived as follows:

$$\Phi(\mathcal{X}) = \sum_{w\in\mathcal{W}} q_w \sum_{i\in\mathcal{F}} p_i u_{i,w} \tag{6}$$

Since $\omega_1 > \omega_0$, minimizing the transmission cost of file download is equivalent to maximizing $\Phi(\mathcal{X})$.

3.2 Distributed Approximate Solution

Before solving the above optimization problem, we first discuss the period of validity T of user requests. The critical value T_c of validity period is defined as

s_{min}/B_{max}. In the case of $T < T_c$, it means that no matter which movement pattern is taken, and no matter which video file is requested, successful recovery of requested videos cannot occur. That is, there are hardly user requests that can be completely served by the local SBSs, which is clearly not what the operator would like to see. Thus, in this paper, we consider that the validity period of user requests satisfies $T \geq T_c$.

Next, we first present the optimal caching strategy when $T = T_c$, and then the iterative optimization on the basis of this result is implemented to maximize the average amount of coded data downloaded from local SBSs, thereby minimizing the transmission cost. In the case of $T = T_c$, Eq. (3) can be simplified as follows:

$$u_{i,w} = \sum_{n \in \mathcal{N}} \min\{x_{n,i}, B_n a_{w,n}\} \tag{7}$$

The Eq. (6) can be rewritten as follows:

$$\Phi(\mathcal{X}) = \sum_{n \in \mathcal{N}} \sum_{w \in \mathcal{W}} q_w \sum_{i \in \mathcal{F}} p_i \cdot \min\{x_{n,i}, B_n a_{w,n}\} \tag{8}$$

Algorithm 1 Mobility-based Optimal Caching Algorithm ($T = T_c$)

Input: B_n, $C_n^{'}$, $\lambda_{n,i}^k$, s_i. **Output**: The optimal solution $\mathcal{X}_n^*$.
$val_{n,i}^k \leftarrow \lambda_{n,i}^k$, $wgt_{n,i}^k \leftarrow s_i$, $i \in \mathcal{F}$, $k \in \{1, \cdots, T\}$; $x_{n,i} \leftarrow 0$, $i \in \mathcal{F}$;
$D \leftarrow \{1, \cdots, F\} \times \{1, \cdots, T\}$;
for $i = 1, 2, \cdots, F \cdot T$ **do**
  while $C_n^{'} > 0$ **do**
    $(i^*, k^*) = \arg\max_{(i,k) \in D} \frac{val_{n,i}^k}{wgt_{n,i}^k}$; $x_{n,i^*} \leftarrow x_{n,i^*} + \min\{B_n, C_n^{'}\}$;
    $D \leftarrow D \setminus (i^*, k^*)$; $C_n^{'} \leftarrow C_n^{'} - B_n$;
  end while
end for

From the structure of $\Phi(\mathcal{X})$ and the aforementioned constraints (5), we can observe that the caching strategy at one SBS does not affect the other SBSs. Therefore, we can decompose this problem into N independent sub-problems and solve them in a distributed way. For SBS_n, the sub-problem $\mathcal{P}_n$ can be expressed as follows:

$$\begin{aligned} &\max_{\mathcal{X}_n} \sum_{w \in \mathcal{W}} q_w \sum_{i \in \mathcal{F}} p_i \cdot \min\{x_{n,i}, B_n a_{w,n}\} \\ &= \sum_{i \in \mathcal{F}} \sum_{k=1}^{T} \sum_{w \in \mathcal{W}_{n,k+}} p_i q_w \cdot d_{n,i}^{(k)} = \sum_{i \in \mathcal{F}} \sum_{k=1}^{T} \lambda_{n,i}^k \cdot d_{n,i}^{(k)} \end{aligned} \tag{9}$$

$$s.t. \quad x_{n,i} \in [0, s_i], \ \forall i \in \mathcal{F}; \quad \sum_{i \in \mathcal{F}} x_{n,i} \leq C_n^{'}. \tag{10}$$

where $\mathcal{W}_{n,k}$ and $\mathcal{W}_{n,k+}$ denotes the subset of movement patterns where the number of time intervals that SBS_n is accessed is equal to k and not less than k, respectively. Thereinto, $\lambda_{n,i}^{k} = \sum_{w \in \mathcal{W}_{n,k+}} p_i q_w$. It can be observed that the objective function in $\mathcal{P}_n$ is a superposition of F monotonically increasing piecewise linear functions.

With respect to the optimization variables $x_{n,i} (i \in \mathcal{F})$, the objective function and the constraints of this problem are linear, and thus it can be solved by using linear optimization techniques. Specifically, we classify problem $\mathcal{P}_n$ as a class of knapsack problems. The optimal solution of this knapsack type problem can be obtained by the following scheme, that is, iteratively placing the item with the highest ratio of value to weight into the knapsack until there is no space left. The specific procedure is summarized in Algorithm 1.

When $T > T_c$, the amount of coded data $u_{i,w}$ downloaded from the SBSs cache can't be reduced to the form of Eq. (7), so we cannot obtain the optimal caching strategy. Thus, an iterative optimization on the basis of the optimal caching strategy $\mathcal{X}_n^*$ is implemented to minimize the transmission cost of cached contents. The criterion of cache optimization is mainly based on the popularity of files. That is, if there are several video files that are stored with the same amount of coded data, then we can consider decreasing the amount of coded data corresponding to the less popular files, while increasing the amount of coded data corresponding to the more popular files. Let $V_{in}(i)$ and $V_{de}(i)$ represent the changes in the average transmission cost $\Omega(\mathcal{X})$ when the amount of cached data of the file f_i is increased or decreased by B_n. When $|V_{in}(i^+)| > V_{de}(i^-)$, the cache optimization can be performed. The specific procedure is described in Algorithm 2.

Algorithm 2 Mobility-based Approximate Caching Algorithm ($T > T_c$)

Input: B_n, $\mathcal{X}_n^*$. **Output**: The approximate solution $\mathcal{X}_n$.
while *true* **do**
  Initialize $\mathcal{F}_{in} \leftarrow \emptyset$, $\mathcal{F}_{de} \leftarrow \emptyset$, $V_{in}(i) \leftarrow 0$, $V_{de}(i) \leftarrow 0$, $i \in \mathcal{F}$;
  $x_0 = \max\{x_{n,i}, i \in \mathcal{F}\}$;
  while $x_0 \geq 0$ **do**
    $i^- = \max\{i | x_{n,i} \geq x_0, i \in \mathcal{F}\}$;
    if $x_{n,i^-} > 0$ **then**
      $\mathcal{F}_{de} \leftarrow \mathcal{F}_{de} \cup \{i^-\}$; calculate the variation in average cost $V_{de}(i^-)$;
    end if
    $i^+ = \min\{i | x_{n,i} \geq x_0, i \in \mathcal{F}\}$; $\mathcal{F}_{in} \leftarrow \mathcal{F}_{in} \cup \{i^+\}$;
    calculate the variation in average cost $V_{in}(i^+)$; $x_0 \leftarrow x_0 - B_n$;
  end while
  if $|\min\{V_{in}(i^+)\}| > \min\{V_{de}(i^-)\}$ **then**
    $i_m^+ = \arg\min_{i^+ \in \mathcal{F}_{in}} V_{in}(i^+)$; $x_{n,i_m^+} \leftarrow \min\{x_{n,i_m^+} + B_n, s_{i_m^+}\}$;
    $i_m^- = \arg\min_{i^- \in \mathcal{F}_{de}} V_{de}(i^-)$; $x_{n,i_m^-} \leftarrow \max\{x_{n,i_m^-} - B_n, 0\}$;
  else
    break;
  end if
end while

4 Chunk-Based Uncoded Caching Strategy

Once a user initiates a video request, the associated SBS immediately informs the MPE to forecast possible movement trajectories within the coverage of the current location. Based on the predicted trajectories and corresponding sojourn time (intervals), the MBS determines the set of SBSs where the cached contents should be updated and the video fragments that they respectively need to store. Upon the user leaves current location and enters the coverage of adjacent location, the above steps are implemented again until the requested file is completely downloaded. The principle of the proposed caching strategy is to store in advance the data chunks in the likely sequences of SBSs that are accessed by the user at a higher probability, and these chunks are most probably downloaded at each small-cell traversed.

The detailed caching scheme is presented as follows. Firstly, the first data chunks of each video file are placed in the SBSs cache at the beginning. The size of cached video fragments of file f_i in the cache of SBS$_n$ is given by $\min\{p_i C_n^{''}, s_i\}$, where $C_n^{''}$ denotes the capacity used to store uncoded video fragments in the SBS$_n$ cache, satisfying $C_n^{''} < C_n$. Secondly, when a video file is requested by the user, the MBS sorts the possible movement trajectories predicted by MPE in descending order of occurrence probability, and then selects the most probable trajectories so that the sum of their probabilities is not less than the threshold τ. Finally, according to the predicted SBSs sequences and the sojourn time in the corresponding small-cell, the data chunks of requested file that should be stored in the cache of each SBS are determined. At the time of performing caching at each SBS, the previous chunk and the latter chunk (if exist) of the decision result should also be placed in the cache, so that the data can also be downloaded from the local cache rather than remote server in the case that the movement speed of the user changes slightly.

Considering that the predicted results cannot be completely accurate no matter what kind of mobility prediction model is adopted, so the several trajectories is chosen instead of the most likely one. Because SBSs have limited storage space, a cache replacement strategy is necessary to determine which data chunks should be removed, in the case that the cache is full and new chunks should be cached. Obviously, the video fragments that have already been delivered are the ones to be evicted.

5 Simulation Results

In this section, we evaluate the performance of the proposed caching strategies. The performance criterion we use is the average transmission cost of the requested files denoted with Ω. In the case of $T \geq T_c$, we compare the performance of the approximate caching strategy in which iterative optimization is implemented with the maximum popularity caching strategy and the optimal caching strategy ($T = T_c$).

In our simulation, we consider that there are $F = 100$ files with a size of 60 MB in the video library, and their popularity follows a Zipf distribution with exponent $\gamma = 1.0$. There are $N = 7$ SBSs with the same cache capacity $C_n^{'} = 20\%$ of the video library size used to store coded data in a macro cell to achieve seamless coverage, and we consider that these SBSs respectively represent different locations. During the movement, the user moves to the adjacent locations with equal transition probabilities. The validity period of user requests and the amount of data delivered by SBS_n are set to $T = 5$ time intervals and $B_n = 20$ MB, respectively. In addition, we assume that the costs of serving a file request from SBSs cache and remote server are 0 and 100.

Impact of Cache Capacity. We first investigate the impact of cache capacity on the average transmission cost incurred by the presented algorithms, as depicted in Fig. 2. In this experiment, the cache capacity of SBSs span a wide range, from 10% to 50% of the entire video files library size. As expected, increasing the cache capacity reduces the average transmission cost. Furthermore, we can observe that the approximate caching algorithm always outperforms the optimal caching algorithm, and they perform significantly better than the maximum popularity caching. This can be explained from the fact that maximum popularity caching policy takes caching decisions considering only user demand, ignoring the movement patterns of users.

Impact of Zipf Exponent. Figure 3 examines the relation between the average transmission cost and the exponent of Zipf distribution γ. We notice that the approximate caching algorithm that performs iterative optimization consistently outperforms the other algorithms, and the gap between their performances diminishes with increasing the parameter of Zipf. As we know, the exponent γ characterizes the correlation level of user requests. The larger the value of it, the more the user preferences are concentrated on a few most popular video files. That is, increasing the value of γ accordingly enhances the probability of requesting files in the SBSs cache, thereby reducing the average transmission cost of the requested contents.

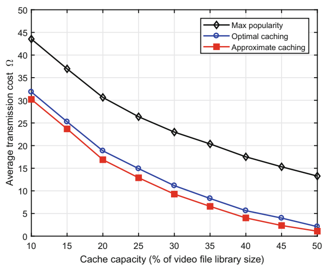

Fig. 2. Average transmission cost Ω versus cache capacity C.

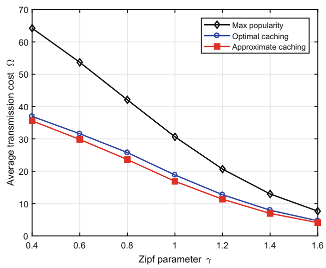

Fig. 3. Average transmission cost Ω versus Zipf exponent γ.

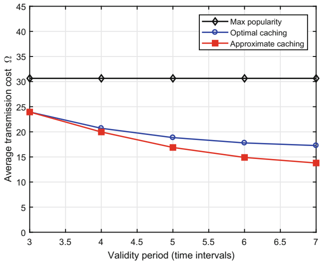

Fig. 4. Average transmission cost Ω versus validity period T.

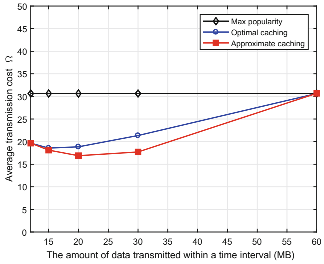

Fig. 5. Average transmission cost Ω versus amount of data delivered within a time interval B.

Impact of Validity Period. We explore how the period of validity of user requests impacts the results in Fig. 4. In this experiment, the validity period T varies from 3 time intervals (corresponding to the critical value T_c) to 7 time intervals. It can be observed that the average transmission cost gradually decreases for the proposed caching algorithms, since the user has more opportunities to contact with different SBSs. The amount of coded data downloaded from the remote server over backhaul link is reduced. It is worth mentioning that the performance of the maximum popularity caching remains unchanged, because the most popular files have been completely replicated in the cache. The change in T has no impact on the amount of data delivered by SBSs.

Impact of Transmission Rate. In Fig. 5, we analyze the impact of the amount of data delivered by SBSs within a time interval on the performance of caching algorithms. Specifically, the transmission rate B varies in $\{12, 15, 20, 30, 60\}$ MB per interval. As the transmission rate increases, it is expected that the SBSs can transmit more data during connection with the user, thus reducing the transmission cost. However, with the data rate continuous increasing, the average cost no longer decreases, but instead starts to increase. In the case of higher transmission rate, we need to store more coded data for each of the most popular files that should be cached. Due to the limited cache capacity, this may cut down the number of requests directly served by the caches.

6 Conclusion

In this paper, due to the different requirements of multimedia applications on user terminals for cached contents, we present two different kinds of caching policies. For ordinary network video files, the caching problem is formulated as an optimization problem to minimize the average transmission cost of cached contents. For typical streaming video, we consider storing uncoded video fragments in the caches. The proposed caching scheme is to cache data chunks in advance at SBSs passed by the user based on the mobility prediction results. The results of simulation reveal that the proposed mobility-based caching performs significantly better than max-popularity caching.

Acknowledgement. This paper is sponsored by the National Science and Technology Major Project of China (Grant No.2017ZX03001014).

References

1. Golrezaei, N., Shanmugam, K., Dimakis, A.G., Molisch, A.F., Caire, G.: Femto-Caching: wireless video content delivery through distributed caching helpers. In: 2012 Proceedings IEEE INFOCOM, March 2012, pp. 1107–1115 (2012)
2. Poularakis, K., Iosifidis, G., Tassiulas, L.: Approximation algorithms for mobile data caching in small cell networks. IEEE Trans. Commun. **62**(10), 3665–3677 (2014)
3. Chae, S.H., Choi, W.: Caching placement in stochastic wireless caching helper networks: channel selection diversity via caching. IEEE Trans. Wirel. Commun. **15**(10), 6626–6637 (2016)
4. Wang, R., Peng, X., Zhang, J., Letaief, K.B.: Mobility-aware caching for content-centric wireless networks: modeling and methodology. IEEE Commun. Mag. **54**(8), 77–83 (2016)
5. Guan, Y., Xiao, Y., Feng, H., Shen, C.C., Cimini, L.J.: MobiCacher: mobility-aware content caching in small-cell networks. In: 2014 IEEE Global Communications Conference, Dec 2014, pp. 4537–4542 (2014)
6. Liu, T., Zhou, S., Tsinghua, Z.N.: Mobility-aware coded-caching scheme for small cell network. In: 2017 IEEE International Conference on Communications, May 2017, pp. 1–6 (2017)
7. Poularakis, K., Tassiulas, L.: Code, cache and deliver on the move: a novel caching paradigm in hyper-dense small-cell networks. IEEE Trans. Mob. Comput. **16**(3), 675–687 (2017)
8. Mishra, S.K., Pandey, P., Arya, P., Jain, A.: Efficient proactive caching in storage constrained 5G small cells. In: 2018 10th International Conference on Communication Systems Networks (COMSNETS), Jan 2018, pp. 291–296 (2018)
9. Cha, M., Kwak, H., Rodriguez, P., Ahn, Y.-Y., Moon, S.: I tube, you tube, everybody tubes: analyzing the world's largest user generated content video system. In: Proceedings of the 7th ACM SIGCOMM Conference on Internet Measurement, 2007, pp. 1–14 (2007)
10. Leong, D., Dimakis, A.G., Ho, T.: Distributed storage allocations. IEEE Trans. Inf. Theory **58**(7), 4733–4752 (2012)
11. Lee, K., Lee, J., Yi, Y., Rhee, I., Chong, S.: Mobile data offloading: how much can wifi deliver? IEEE/ACM Trans. Netw. **21**(2), 536–550 (2013)