An Edge Caching Strategy for Minimizing User Download Delay

Tianhao Wu, 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
{wutianhao,lixi,jihong,zhangheli}@bupt.edu.cn

Abstract. In the scenario with many small base stations (SBSs) deployed, edge caching technology could bring contents closer to users by caching files at SBSs. Considering these SBSs with limited storage capacity, how to effectively cache files is a difficult and interesting problem. Many factors should be considered, such as the popularity of files, user download delay and average hit rate. In this paper, we investigate this problem and propose a minimizing user download delay caching (MUDDC) algorithm. It decides which files should be cached and where to cache them for reducing download delay. There is a conflict between hit rate and download delay with the limited SBS storage capacity. We target the average user download delay and model an optimization problem with the constraint of average hit rate and find the optimal solution. The simulation results show that the system performance is improved.

Keywords: Edge caching network · Content distribution
Download delay · Hit rate

1 Introduction

As the explosive growth of smart devices and the advent of many new applications, traffic volume has been growing exponentially [1]. With the rapid growth of traffic demands in future cellular networks, one promising approach is to deploy more small base stations (SBSs) along with macro base stations (MBSs) [2]. In order to deal with the data requirements, the contents can be cached in SBSs, bringing the files closer to users [3]. It reduces the duplicate content transmissions and allows users to acquire files with less download delay. However, the storage capacity of these SBSs is limited. In this case, one SBS could not cache all the files that users may need. So some studies concentrate on a specific SBS storage mode in this circumstance. The cached files could be divided into two

T. Wu—This work is supported by National Natural Science Foundation of China under grant 61671088 and National Science and Technology Major Project of the Ministry of Science and Technology of China under grant 2016ZX03001017.

L. Wang et al. (Eds.): QShine 2017, LNICST 234, pp. 120–130, 2018.
https://doi.org/10.1007/978-3-319-78078-8_13

parts. The first part has higher popularity and be stored by all the SBSs. The second part has lower popularity and are distributed in different SBSs. It is a kind of practical approach for edge caching.

There are many aspects should be considered when designing a caching scheme. With the limited storage, how to decide whether to store a file or not in a SBS connects with not only the file's popularity and hit rate, but also with users experience like transmission delay and personal preference. Moreover, spectrum resource, traffic offloading and throughput may also be taken into account.

In [4], the authors study the problem of content placement for caching at the wireless edge with the goal to maximize the energy efficiency (EE) of heterogeneous wireless networks. In [5], the authors introduce the optimal edge caching strategies to minimize the bandwidth consumption of fronthaul and storage costs in the fog radio access networks (F-RANs). In [6], the authors formulate an optimal redundancy caching problem to minimize the total transmission cost of the network, including cost within the radio access network (RAN) and cost incurred by transmission to the core network via backhaul links. In [7], the authors design the joint caching and association strategy to minimize the average download delay in a cache-enabled heterogeneous network. In [8], the authors propose a novel edge caching scheme to cache layered contents. The proposed method outperforms the exiting counterparts with a higher hit ratio and lower delay of delivering video contents.

Despite the previous work on edge caching network, it still needs further study that jointly consider the hit rate and user download delay. Because these two basic factors are significant indicators in evaluating network performance. Moreover, in the network scenario with limited storage, the growth of the hit rate may be conflict with the user average download delay. It is necessary to achieve a tradeoff between them.

In this paper, we study the edge caching scheme to minimize user average download delay under the premise of reaching a certain average hit rate and propose a minimizing user download delay caching (MUDDC) algorithm. The set of files stored in each SBS could be established according to a certain file placement strategy. All the files are sorted according to their popularity rank. A threshold value R is set to divide the stored files into two parts. The most popular R files are stored in every SBS to reduce average user download delay. While the rest files are distributed stored by different SBSs according to their storage capacity. This increases the diversity of stored files and improves the average hit rate to some extent. In this network scenario, the proposed algorithm establishes a connection state matrix for users and SBSs. In above conditions, the user average hit rate and download delay could be calculated. Eventually, we model the tradeoff between hit rate and delay into an optimization problem and find the optimal file placement strategy. Simulations are conducted which show that the proposed algorithm has a low average download delay and could achieve a good hit rate. Furthermore, the proposed algorithm could significantly improve the average hit rate with the similar download delay compared to that of the "most popular" placement scheme.

The remainder of the paper is organized as follows. Section 2 gives the system model and problem formulation. Section 3 indicates the whole process of MUDDC algorithm. Simulation results and discussions are given in Sect. 4. Finally, we conclude this paper in Sect. 5.

2 System Model and Problem Formulation

2.1 System Model

The edge caching network considered in this paper is shown in Fig. 1. The network is composed by one MBS and N SBSs with J mobile users in their coverage areas. Each SBS stores files according to the file placement strategy. For different files, the sizes of them are same. In Fig. 1, each colored square means one file and different colors denote different files.

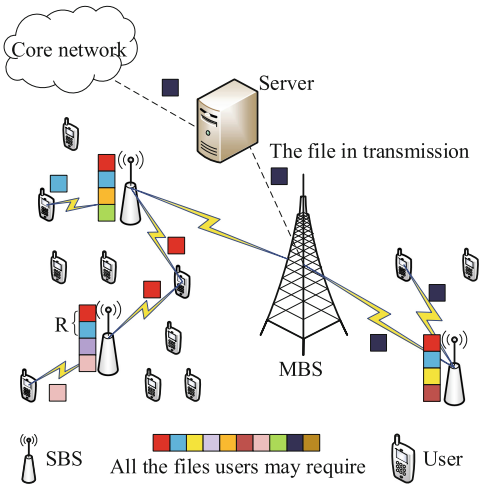

Fig. 1. Edge caching network (Color figure online)

The MBSs connect to the core network with wire links. The SBSs and MBSs are connected with wireless links. The users connect to SBSs via wireless links. The coverage areas of the SBSs may be sometimes overlapping, and therefore users can potentially be served by multiple SBSs. When a user wants to get a file from the network, the user requires contents from the storage of the SBSs they connecting to at first. When all the SBSs have this file, the user will control the SBSs it connects to cooperate with each other for getting a faster download speed. For example, the download speed could be nearly double with the user connecting to two SBSs. If only one SBS stores this file, the download speed is normal. In the circumstance that the file is not in the storage, one of the SBSs would require for the core network through the MBS and send to the user.

We assume that the set of N SBSs is $\mathbb{N} = \{1, 2, \cdots, n, \cdots, N\}$, where n is the nth SBS. We denote the set of J mobile users by $\mathbb{U} = \{U_1, U_2, \cdots, U_j, \cdots, U_J\}$, where U_j represents the jth user. For the jth user U_j, there is a matrix $\mathbb{A}_j$ which means the connectivity condition between U_j and the SBSs. The expression of $\mathbb{A}_j$ is $\mathbb{A}_j = [a_j^1, a_j^2, \cdots, a_j^n, \cdots, a_j^N]$, where a_j^n is a binary-state variable. Its value is either 0 or 1. When user j has connected to SBS n, the value of a_j^n is 1, otherwise the value is 0.

2.2 SBS Caching Model

The way that SBSs store files proposed here refers to [6] and is shown in Fig. 2. A SBS could store M files at most. We assume that the mobile users may request a file from Y ($Y \geq M \cdot N$) different popular files at a time. These files are predicted by the statistics of user information. All the Y files in the system are sorted according to their popularity rank. Each file corresponds to a serial number k, and the value of k is smaller means the file is more popular. As shown in Fig. 2, The first R files cached in all the SBSs are the most popular R files in the system. The remaining m ($m = M - R$) files cached in each SBS are different from others and are stored according to their serial numbers [6].

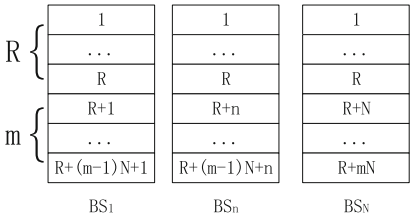

Fig. 2. The caching scheme

For the nth SBS, we could establish a set $\mathbb{K}_n$ to denote the files stored in this SBS. So $\mathbb{K}_n$ could be expressed as:

$$\begin{aligned}\mathbb{K}_n = \{&1, 2, \cdots, R, R+n, R+N+n, \\ &\cdots, R+(m-1)N+n\}.\end{aligned} \tag{1}$$

The contents of the collection are the serial numbers of the stored files in the nth SBS.

2.3 Average Hit Rate Model

From $\mathbb{K}_n$, we would know which files are stored in the nth SBS. So a matrix $\mathbb{H}_n$ could be established to reflect the relationship between the nth SBS and all the

Y files. The $\mathbb{H}_n$ would be denoted as:

$$\mathbb{H}_n = [h(1), h(2), \cdots, h(k), \cdots, h(Y)],$$
$$h(k) = \begin{cases} 1 & k \in \mathbb{K}_n \\ 0 & others \end{cases}. \tag{2}$$

The value of $h(k)$ is 1 means the kth file is stored in the nth SBS, otherwise the value is 0.

A user j could directly get files stored in the SBSs it connects to. So we would get a set $\mathbb{F}_j$ to express these files for the user. The expression of $\mathbb{F}_j$ is

$$\mathbb{F}_j = a_j^1 \cdot \mathbb{K}_1 \cup a_j^2 \cdot \mathbb{K}_2 \cup \cdots \cup a_j^N \cdot \mathbb{K}_N. \tag{3}$$

The elements of $\mathbb{F}_j$ are the serial numbers of the files user j could acquire from the SBSs directly.

According to the statistical result, a user has a probability P_k to require the kth file from the SBSs. From [6], we can get P_k through

$$P_k = \frac{1/k^\beta}{\sum_{y=1}^{Y} (1/y^\beta)}. \tag{4}$$

In (4), β is a decay constant. So when the user j connecting to the SBSs, the expected hit rate is calculated as

$$q_j = \sum_{k \in \mathbb{F}_j} P_k. \tag{5}$$

After known q_j for every user, we could get the average hit rate Q of the network:

$$Q = \frac{\sum_{j=1}^{J} q_j}{J}. \tag{6}$$

2.4 Average User Download Delay Model

The SBSs reuse the downlink resources of the MBS to serve the transmission to users [7]. As a result, there exists the interference by the MBS when users obtain files from SBSs. The neighboring SBSs could be allocated orthogonal frequency band to eliminate the in-layer interference. Each SBS divides its downlink bandwidth into many subchannels. All the subchannels have the same bandwidth ω_n. Each user accesses only one subchannel at a slot. Similarly, each subchannel of the MBS has the bandwidth ω_M.

Let P_n be the transmission power of the SBSs, and P_M is the transmission power of the MBS. Denote the noise power as σ^2. The channel gain between SBS n and user j could be $h_{n,j}$. Also $h_{M,n}$ means the channel gain between the MBS

and SBS n. As in [9], we characterize the channel gain model as $h_{n,j} = L_0 d_{n,j}^{-\alpha}$ and $h_{M,n} = L_0 d_{M,n}^{-\alpha}$, where L_0 is a constant and α is the path loss exponent factor. $d_{n,j}$ and $d_{M,n}$ are the distance between SBS n and user j and the distance between MBS and SBS n respectively.

Therefore, the signal-to-interference-plus-noise ratio (SINR) $\gamma_{n,j}$ between SBS n and user j and the SINR $\gamma_{M,n}$ between MBS and SBS n could be denoted as $\gamma_{n,j} = \frac{P_n h_{n,j}}{\sigma^2 + P_M h_{M,n}}$ and $\gamma_{M,n} = \frac{P_M h_{M,n}}{\sigma^2}$. We assume that the size of all files is L. Thus we would know the delay ($D_{j,n}$) of SBS n sending a file to user j and the delay ($D_{n,M}$) of SBS n receiving a file from MBS. So $D_{j,n}$ could be represented as

$$D_{j,n} = \frac{L}{\omega_n \log_2(1 + \gamma_{n,j})}, \tag{7}$$

and $D_{n,M}$ is represented as

$$D_{n,M} = \frac{L}{\omega_M \log_2(1 + \gamma_{M,n})}. \tag{8}$$

2.5 Problem Formulation

In order to acquire the file download delay of every user, we would establish a file available degree matrix $\mathbb{T}_j$ for user j. The form of $\mathbb{T}_j$ could be $\mathbb{T}_j = [t_j(1), t_j(2), \cdots, t_j(k), \cdots, t_j(Y)]$, where $t_j(k)$ means the number of SBSs stored the kth file in connection with user j and $\mathbb{T}_j$ is calculated by

$$\mathbb{T}_j = \sum_{n=1}^{N} a_j^n \cdot \mathbb{H}_n. \tag{9}$$

The user download delay may be various with different files, so we should calculate the download delay of every file for user j. Thus, the download delay $D_j(k)$ of user j receiving file k is denoted as:

$$D_j(k) = \begin{cases} \frac{1}{t_j(k)} \cdot D_{j,n} & t_j(k) \geq 1 \\ D_{j,n} + D_{n,M} & t_j(k) = 0 \end{cases}. \tag{10}$$

Then the average file download delay D_j of user j would be calculated with

$$D_j = \sum_{k=1}^{Y} P_k \cdot D_j(k). \tag{11}$$

Therefore, the average user download delay $D(R)$ of the system could be known after calculating the delay of every user. The expression of $D(R)$ is

$$D(R) = \frac{\sum_{j=1}^{J} D_j}{J}. \tag{12}$$

From above analysis, we have expressed the average hit rate Q and the average user download delay $D(R)$. Our goal is to find an optimal file placement strategy for SBSs to minimize $D(R)$ under the premise of reaching a certain average hit rate Q_c. Considering that the value of R would not be bigger than the SBS storage capacity M, so the optimization problem could be modeled as

$$\begin{aligned} &\min D(R) \\ \texttt{s.t.}\quad &Q \geq Q_c \\ &R \in \mathbb{Z} \\ &0 \leq R \leq M \end{aligned} \tag{13}$$

From (13), we know that R is an integer and $0 \leq R \leq M$. The traversal algorithm should be a useful way to find the optimum value of R.

3 The Proposed MUDDC Algorithm

The MUDDC algorithm proposed in this paper is divided into two parts. The first part is the process of SBS file distribution strategy. The second part is the process of user connection and requiring files.

For SBSs, the connectivity condition of all users should be determined. In the edge caching network, a user could connect to many SBSs. So the connectivity condition matrix for all the users and SBSs should be established firstly.

For users, the popular files that they may require should be predicted and ranked. The serial number is smaller means the file is more popular. Next each requiring probability of the files can be calculated by (4).

In order to find the optimal value of R conveniently, we should calculate $D_{j,n}$ and $D_{n,M}$ in advance by using (7)–(8).

When all the initial conditions are decided, the MUDDC algorithm will use (13) to model an optimization problem and use the traversal algorithm to solve the problem. After finding the appropriate strategy of file placement mode, the SBSs could be deployed with this strategy. All the SBSs should update their stored files simultaneously, and have the same way to store files. Then all the users begin the process of requiring files.

When the connected user wants to receive a file, it will firstly check whether this file has been stored in the SBSs. If this file is stored in all SBSs, the user will let connected SBSs collaborate with each other to shorten download delay. When only one connected SBS has this file exactly, this SBS sends the required file to the user. The download delay is normal. If this file is not stored in any connected SBSs, one of the SBSs would require the file from MBS and then send to the user. This would add delay of MBS sending the file to SBS to download delay. The process of the MUDDC algorithm is summarized in Algorithm 1.

4 Simulation Results and Discussions

In this section, we evaluate the performance of the MUDDC algorithm and analyze the influence of important parameters by simulation. For comparison,

Algorithm 1. The MUDDC Algorithm

1: The users connect to the SBSs.
2: Initialization:
a)Set $\mathbb{N}$, $\mathbb{U}$, and matrix $\mathbb{A}_j$;
b)SBS storage capacity M, the number of all files Y, the size of the files L, and the threshold value of hit rate Q_c;
c)SBS transmission power P_n, transmission power P_M, noise power σ^2, channel gain h_n and h_M.
3: Model the optimization problem by (1)–(13).
4: Use traversal algorithm to find the value of R which minimizes the delay $D(R)$.
5: SBSs execute the file placement strategy shown in Fig. 2 according to R.
6: Users begin to require files.
7: **for all** $U_j \in \mathbb{U}$ **do**
8: User U_j requires a file.
9: **if** this file is stored in all SBSs **then**
10: The connected SBSs would cooperate to send this file to U_j.
11: **else**
12: **if** this file is stored in only one SBS **then**
13: The stored SBS sends the file to U_j.
14: **else**
15: One of the connected SBS require the file from MBS and then send to U_j.
16: **end if**
17: **end if**
18: **end for**

we emulate the "most popular" placement (MPP) scheme mentioned in [4]. In MPP scheme, each SBS stores the M most popular files.

In the simulation, we let the size of each file $L = 10$ Mbits, $\beta = 0.8$, $Y = 600$, $M = 50$, $J = 50$, and $\alpha = 4$. In [4], we know that the MBS coverage area radius is 500 m, and the SBS coverage area radius is 45–60 m. For simplicity, we assume $d_{n,j} = 30$ m for all SBSs and users and $d_{M,n} = 200$ m for all SBSs and the MBS. The values of P_n and P_M are 100 mW and 20 W respectively, and $L_0 = -30$ dB [9]. The value of σ^2 is -100 dBm [10]. We assume that $\omega_M = \omega_n = 1$ MHz.

Figure 3 shows the relationship between average user download delay and R on the condition of different value of Q_c. The value of the threshold Q_c is related to the system performance and has a certain randomness. In Fig. 3, from 28 to 46 the value of R, the value of Q_c is $0.525, 0.520, 0.515, 0.510, 0.505, 0.500, 0.495$ respectively. The number of SBSs is 4. From Fig. 3, we know that reducing the value of R is conducive to improving user hit rate. And this would increase the download delay. The larger the value of R is, the more files there are stored in all the SBSs. Therefore the download delay of these most popular files is shortened, and then decreases the average download delay. However, the file diversity would be lower. Thus the poor average hit rate appears. This is the reason of the conflict between these two factors.

Figure 4 indicates the changes of R with different number of total SBSs. We set the value of Q_c according to the system performance. In Fig. 4, from 4 to 10

the total number of SBSs, the value of Q_c is $0.525, 0.53, 0.51, 0.51, 0.505, 0.5, 0.46$ respectively. It is the same with Figs. 5 and 6. When the total number of SBSs is increasing, the files would be placed more dispersedly. The hit rate decreases with the reduced value of R. Moreover, the download delay may be increased. So the files stored in all SBSs is more with the number of total SBSs to improve the system performance.

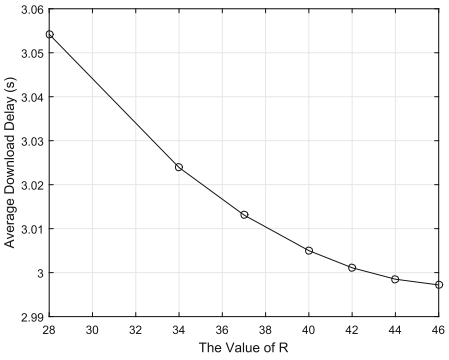

Fig. 3. Average download delay with different value of R

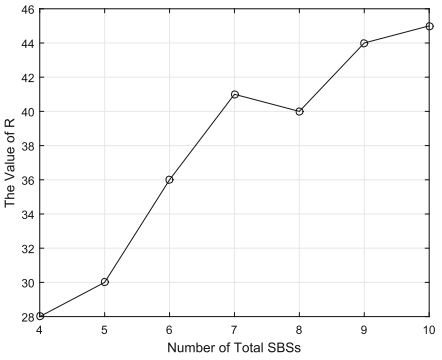

Fig. 4. The value of R with different number of total SBSs

In Fig. 5, we compare the performance of average hit rate between the MUDDC algorithm and the MPP algorithm. From Fig. 5, the network performance of the MUDDC algorithm is obviously better than the MPP algorithm in the matter of average hit rate. That is because the MPP algorithm let all the SBSs store M most popular files. So no matter how many SBSs a user connects

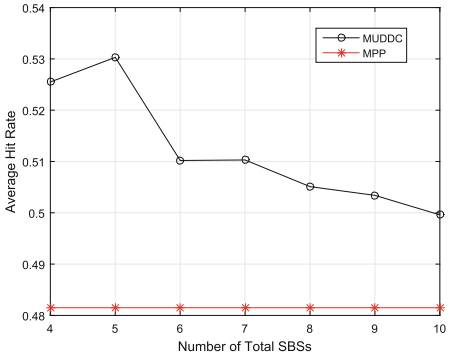

Fig. 5. The average hit rate with different number of total SBSs

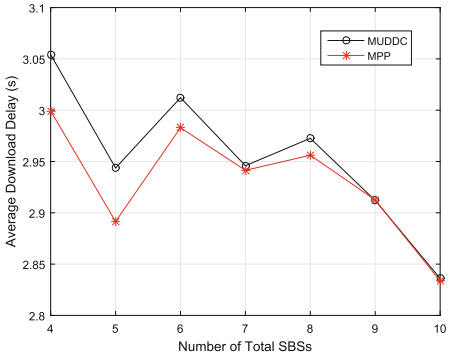

Fig. 6. The average download delay with different number of total SBSs

to, only M files could be acquired directly from SBSs. The average hit rate is not changed. However, all the SBSs store R files and $R \leq M$. This may make a user get more than M files from the connected SBSs directly. As a result, the average hit rate in MUDDC algorithm is higher than that in MPP algorithm.

Figure 6 illustrates the relationship between the average download delay and the number of total SBSs in above two algorithms. In general, the download delay decreases as the increasing number of SBSs. This is because the value of R has a rising trend. Figure 6 clearly shows that there is little difference on average download delay between MUDDC and MPP algorithm. However, the average hit rate in MUDDC algorithm is higher than MPP compared with Fig. 5. This illustrates the proposed MUDDC algorithm successfully minimizes the download delay under the condition of achieving an average hit rate Q_c to some extent.

5 Conclusion

In this paper, the MUDDC algorithm, a file placement algorithm for limited SBSs storage has been proposed. On the premise of meeting a certain average hit rate, the MUDDC algorithm could minimize the download delay. In this algorithm, an optimization model is established and a traversal algorithm is used to find the optimal solution. The simulation results show that compared to the existing algorithm, the MUDDC algorithm could reach a higher hit rate with the similar download delay. For future work, to improve the system performance, we would consider the MUDDC algorithm with social features and derive the optimal solution for the problem.

References

1. Wang, S., Zhang, X., Zhang, Y., Wang, L., Yang, J., Wang, W.: A survey on mobile edge networks: convergence of computing, caching and communications. IEEE Access **5**, 6757–6779 (2017)
2. Ghosh, A., Mangalvedhe, N., Ratasuk, R., Mondal, B., Cudak, M., Visotsky, E., Thomas, T.A., Andrews, J.G., Xia, P., Jo, H.S., Dhillon, H.S., Novlan, T.D.: Heterogeneous cellular networks: from theory to practice. IEEE Commun. Mag. **50**(6), 54–64 (2012)
3. Yang, C., Yao, Y., Chen, Z., Xia, B.: Analysis on cache-enabled wireless heterogeneous networks. IEEE Trans. Wirel. Commun. **15**(1), 131–145 (2016)
4. Gabry, F., Bioglio, V., Land, I.: On energy-efficient edge caching in heterogeneous networks. IEEE J. Sel. Areas Commun. **34**(12), 3288–3298 (2016)
5. Wang, X., Leng, S., Yang, K.: Social-aware edge caching in fog radio access networks. IEEE Access **5**, 8492–8501 (2017)
6. Wang, S., Zhang, X., Yang, K., Wang, L., Wang, W.: Distributed edge caching scheme considering the tradeoff between the diversity and redundancy of cached content. In: 2015 IEEE/CIC International Conference on Communications in China (ICCC), pp. 1–5, November 2015
7. Wang, Y., Tao, X., Zhang, X., Mao, G.: Joint caching placement and user association for minimizing user download delay. IEEE Access **4**, 8625–8633 (2016)

8. Su, Z., Xu, Q., Hou, F., Yang, Q., Qi, Q.: Edge caching for layered video contents in mobile social networks. IEEE Trans. Multimedia **19**(10), 2210–2221 (2017)
9. Jia, C., Lim, T.J.: Resource partitioning and user association with sleep-mode base stations in heterogeneous cellular networks. IEEE Trans. Wirel. Commun. **14**(7), 3780–3793 (2015)
10. Zhang, H., Wang, Y., Ji, H.: Resource optimization-based interference management for hybrid self-organized small-cell network. IEEE Trans. Veh. Technol. **65**(2), 936–946 (2016)