VNU Journal of Science:
Computer Science and Communication Engineering

Journal homepage: http://www.jcsce.vnu.edu.vn/index.php/jcsce

Original Article

Cooperative Caching in Two-Layer Hierarchical Cache-aided Systems

Hoang Van Xiem[1], Duong Thi Hang [1,2], Trinh Anh Vu[1,*], Vu Xuan Thang[3]

[1]*VNU University of Engineering and Technology,144 Xuan Thuy, Cau Giay, Hanoi, Vietnam*
[2]*Hanoi University of Industry, 298 Cau Dien, Minh Khai, Bac Tu Liem, Hanoi, Vietnam*
[3]*Interdisciplinary Centre for Security, Realiability and Trust (SnT) - University of Luxembourg, 2, avenue de l'Université, 4365 Esch-sur-Alzette, Luxembourg*

Received 29 November 2018
Revised 04 March 2019; Accepted 15 March 2019

Abstract: Caching has received much attention as a promising technique to overcome high data rate and stringent latency requirements in the future wireless networks. The premise of caching technique is to prefetch most popular contents closer to end users in local cache of edge nodes, e.g., base station (BS). When a user requests a content that is available in the cache, it can be served directly without being sent from the core network. In this paper, we investigate the performance of hierarchical caching systems, in which both BS and end users are equipped with a storage memory. In particular, we propose a novel cooperative caching scheme that jointly optimizes the content placement at the BS's and users' caches. The proposed caching scheme is analytically shown to achieve a larger global caching gain than the reference in both uncoded - and coded caching strategies. Finally, numerical results are presented to demonstrate the effectiveness of our proposed caching algorithm.

Keywords: Hierarchical caching system, cooperative caching, caching gain, uncoded caching, coded caching.

1. Introduction

Among potential enabling technologies to tackle with stringent latency and data hungry requirements in future wireless networks, edge caching has received much attention [1]. The basic premise of edge caching is to bring the content closer to end users via distributed storages at the edge network. Caching usually comprises a placement phase and a delivery phase. The former is executed during off-peak hours when the network resources are abundant, in which popular content is prefetched in the distributed caches. The later usually occurs during peak-hours when the content requests are revealed. If the requested content is already available in the edge node's local cache, it can be served directly without being sent from the core network. In this manner, edge caching not only leverages backhaul traffic but also reduces

* Corresponding author.
E-mail address: vuta@vnu.edu.vn

https://doi.org/10.25073/2588-1159/vnuer.222

transmission latency significantly, thus mitigating network congestion [2, 3].

The caching technique is usually divided into two types: uncoded and coded caching. In the former, the placement and delivery phases in one cache are independent from others. On the other hand, the later requires cooperation among the caches in both placement and delivery phases. As a result, the coded caching strategy achives a global caching gain in addition to local caching gain of the uncoded scheme.

The investigation of the coded caching has received much attention recently. In [3], the authors studied the caching system under uncoded prefetching while the authors in [4-6] analyzed the coded caching under realistic assumptions by considering a nonuniform distribution of content demands. The impacts of caching in interference networks have been analyzed in both traffic reduction and latency minimization [7-11]. In addition, emerging issues related to distributed caching, caching online were studied in [4, 12, 13, 14, 17]. Especially, the authors in [2, 15] consider a two-layer hierarchical communication system with a server to be connected with end users through a BS. This structure can be extended into multi-level communication system which is able to combine the power of computer and communication systems in 5G. Extention to multiple-server scenario is studied in [10, 16] subject to the total power constrains.

In this paper, we propose a novel cooperative caching scheme at BS and users to reduce the backhaul traffic; hence, improving the overall caching efficiency. Comparing with the work in [2] in which the cache placement in the BS is independent from the users, our scheme jointly optimizes the placement phase at the BS and users. Especially, when the transmission load on access line is added with some unicast message, the additional overall gain on the backhaul line can be achieved.

The organization of this paper is as follows. Section 2 presents the background works on Coded Caching with the global and local gains. After that, Section 3 presents system model and proposes a two-level communication structure with the joint BS and user co-operation. Section 4 examines the proposed caching solution with several scenarios. Finally, Section 5 gives some conclusions and future works.

2. Background works on coded caching

We consider a basic communication system with the following components: a data center containing N files of content, the size of each file is Q (bits). K users can access to the data center through a common line as shown in Fig. 1.

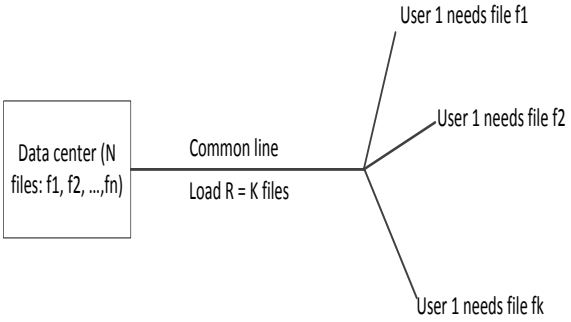

Fig. 1. Basic communication system.

If a user requests a different file from the data center, the maximum of the transmission load (R) on the common line will be:

$$R = K \text{ (files) or } R = K \times Q \text{ (bits)} \qquad (1)$$

Besides, if users request files with similar content, the transmission load will be reduced as the data center can broadcast files.

We consider the case that a user has a memory size of M (files) (0<M<N). To satisfy the requirements from users, two phases can be performed:

Placement phase (or caching phase):

This phase sends a part of content from the data center to users. This happens when the common line is free and there is no specific request from users.

Delivery phase:

This phase is performed when there is a specific request from users. If the required content is already available at the user, it can be

directly extracted. Otherwise, a request will be sent to the data center for the missing part of content.

Local gain

In the placement phase, a part of data content is prefetched into the memory of users as shown in Fig. 2.

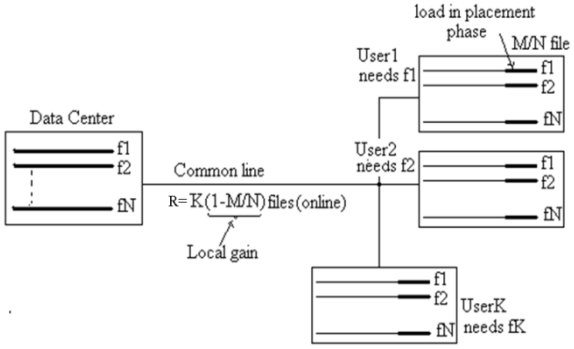

Fig.2. Local gain achievement with placement phase.

In this case, an equal part of the data content is prefetched into the memory of users which will take about M/N files to guarantee that with N files, the size of memory will be M.

As it can be seen, when there is a specific request, the transmission load will be the maximum if each user requires a different content. Because each user needs a missing part of content from the data center the transmission load will be:

$$R = K\left(1 - M/N\right) \text{ (files)} \qquad (2)$$

Comparing to (1), the transmission load is reduced with a factor of (1-M/N). This factor depends on the size of the memory M. this is called the local gain. The above relationship can be depicted as in Fig. 3.

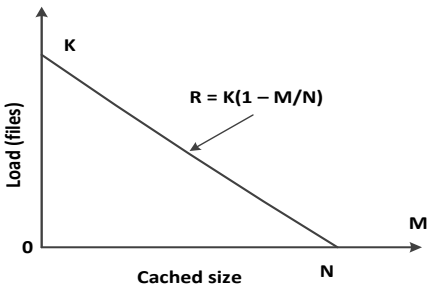

Fig. 3. The relationship between the transmission load and the size of cache memory at the user.

If M=N, the transmission load will be zero. It means that only cache memory is enough for any request.

In summary, the cache memory at the user plays an important role in reducing the transmission load. This is the conventional caching technique or uncoded caching which has been popularly known with the computer architecture.

B. Global gain

The communication system shown in Fig.1 is again considered. However, the placement and delivery strategies are changed as in the following [3]:

For easy tracking, we consider the case with N=K=3 and M=1. Three files from the data center are A, B, C and each file is devided into 3 equal part A=(A1, A2, A3), B=(B1,B2,B3),

C=(C1,C2,C3). The prefetching strategy into memory Zk of user k will be performed as: Zk=(Ak,Bk,Ck) where k=1,2,3 as shown in Fig. 4. It should be noted that by using this caching strategy, the sum of content parts of a file pre-fetched on all users are completely coverted to this file.

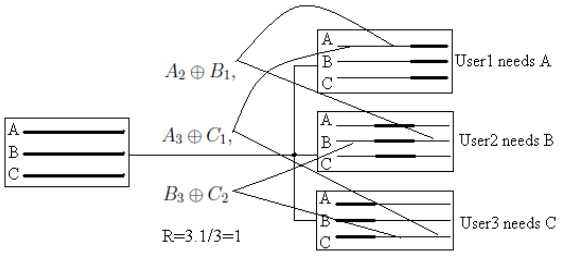

Fig. 4. The placemet and deliver multicast strategy.

When user 1, 2, 3 request 3 different files A, B, C, the data center needs only to send three messages A2⊕B1,A3⊕C1, B3⊕C2 on the common line. Two first messages help the user1 reconstructing A2, A3 because B1, C1 already have in data center. The messages 2 and 3 help the user 3 reconstructing C1, C2 since it already has A3, B3. Finally, the message 1 and 3 help the user 2 reconstructing B1, B3 since it already has A2, C2. As the size of each message is 1/3 file, the transmission load will be R = 3×1/3 = 1(file). This technique is called

coded caching which was proposed by M. Ali et al. in 2014 [3]. Compared to the conventional uncoded caching technique with transmission load is:

$$R = K\left(1 - M/N\right) = 3\times(1\text{-}1/3) = 2 \text{ (files)} \quad (3)$$

It is clearly that the coded caching has significantly reduced the transmission load by using a common message for 2 users as presented above. In this case, the A2⊕B1 used for user 1 and user 2 is called multicast message.

In [3], Generalization for any communication system with N files, K users, and M is the size of memory, (0<M<N), the coded caching technique can be summarized as: Let m = KM/N (0<m<K) and to simply assume that m is an integer (when M ∈ {0, 2N/K,…, N},). Dividing each file f_n at the center into C_K^m equal parts. The size of each part will be Q/C_K^m

Given the index for part is $f_{n,\tau}$ as follow

$f_n = \left(f_{n,\tau} : \tau, \tau \in [K], |\tau| = m\right)$ for m specific users.

In placement phase, prefetching $f_{n,\tau}$ into the cache of user which belongs to τ (a set of m specificed users). Each user will cache NC_{K-1}^{m-1} parts; hence, with m = KM/N, the memory condition will be satisfied

$$\frac{NC_{K-1}^{m-1}Q}{C_K^m} = MQ \quad (4)$$

In delivery phase, if user k requests file f_{dk}, for each set τ containing m users, the data center will send the message $\oplus_{k\subset\tau} f_{dk,\tau\setminus\{k\}}$, which is the sum of modulo-2 of related parts having the same size of Q/C_K^m bits over the common line. There are C_K^{m+1} message used to satisfy all requests. Therefore, the overall transmission load will be:

$$R = \frac{C_K^{m+1}\times Q}{C_K^m} = Q\frac{K-m}{m+1} \quad (5)$$

with $m = kM/N$ we have:

$$R = QK\left(1 - M/N\right)\frac{1}{1+KM/N} \quad (6)$$

Comparing to (2), normalized with the size of file is Q, the additional gain of the transmission load is 1/(1+K×M/N); this is called global gain in which a message will be the multicast message for 1+K×M/N users.

Fig. 5 illustrates the global gain and local gain for uncode and coded caching solutions.

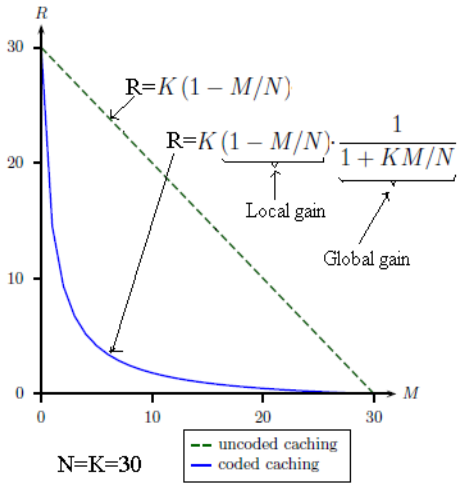

Fig.5. Coded caching và uncoded caching comparison.

Fig.5 shown that if N=K=30, and M=10, the coded caching technique is able to reduce the transmission load 11 times than the uncoded caching technique [3].

3. System model

We consider a two-layer hierarchical caching system which consists of one data center, one BS and K users, as depicted in Fig. 6. The BS serves the users via (access) channels and connects with the data center via a backhaul line. The data center contains a libarary of N contents of equal size of Q bits. Both the BS and users are equipped with a storage memory with the size of Mb and Mu (files), respectively, where 0< Mb, Mu < N. We consider both uncoded and coded caching strategies.

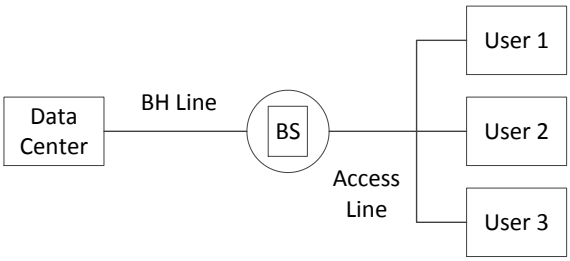

Fig. 6. Two – layer hierarchical caching system.

At the same time, assuming that the transmission line is error free and delay free. This model is different model in [3] when there is more BS and its memory. For this communication system with 0<Mb, Mu<N the authors in [2] have computed the transmission load as:

- For uncoded caching case:

$$Q_{unc.AC} = KQ\left(1-\frac{M_u}{N}\right) \tag{7}$$

$$Q_{unc.BH} = KQ\left(1-\frac{M_u}{N}\right)\left(1-\frac{M_b}{N}\right) \tag{8}$$

Here $Q_{unc.AC}, Q_{unc.BH}$ are the transmission load on the user's access line and on backhaul (BH).

- For coded caching case:

$$Q_{cod.AC} = (1-\delta)\frac{Q(K-m)}{m+1} + \delta\frac{Q(K-m-1)}{m+2} \tag{9}$$

$$Q_{cod.AC} = (1-\delta)\left(1-\left(\frac{M_b}{N}\right)^{m+1}\right)\frac{Q(K-m)}{m+1} + \delta\left(1-\left(\frac{M_n}{N}\right)^{m+2}\right) \tag{10}$$

Here: $m = \left|\frac{kM_u}{N}\right| \in Z^*$ and $\delta = \frac{kM_u}{N} - m$ with $0 \le m < 1$

When KM_u is divisible for N, δ=0, we have:

$$Q_{code,BH} = \left[1-\left(\frac{M_b}{N}\right)^{m+1}\right]\frac{Q(K-m)}{m+1} \tag{11}$$

The results in (8), (9) reveal that adding a memory with size of Mb for BS has reduced the transmision load with a factor of (1-Mb/N) for uncoded caching case and $(1-(Mb/N)^{m+1})$ for coded caching case. These results are based on an assumption that the content stored in Mb is indepent with the content stored in Mu. Concretely, the probability of one-bit content of one file stored in Mb is (Mb/N), hence, the probablity for this bit sending in the transmission line is (1-Mb/N). When having Mb, the overall transmission load will be reduced with a factor of (1-Mb/N). For coded caching, since each bit in the multicast message on access line is XOR of bits from (m + 1) different files. Due to prefetching is independent, the probability for the availabililty of this bit in cache Mb is $(Mb/N)^{m+1}$, thus, the probability for this bit on the backhaul line is $1-(Mb/N)^{m+1}$. This is indeed the reduction factor for the case of BS with additional cache Mb.

When cooperating the content storage between Mb and Mu, we achieve the following results:

Proposition 1:

For a communication system with N, K, Mb, Mu (Mb+Mu<N), when the prefetched content is cooperated between Mb and Mu, The transmission load in backhaul line for uncoded caching is:

$$Q_{unc,BHj} = KQ\left(1-\frac{M_u+M_b}{N}\right) \qquad (12)$$

Proof:

Fig.7 shows the proposed BS and user cooperation in using cache memory.

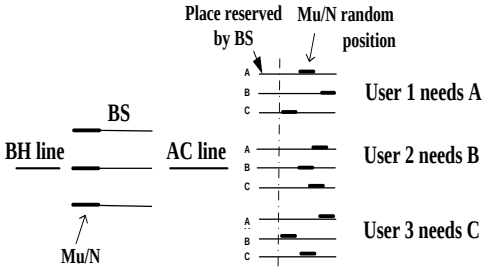

Fig. 7. The cooperated caching between Mb and Mu.

At the placement phase, M_b of BS is used to store the first part of files in the data center. The size of this part is M_b/N. While M_u of users will be used to store (can be randomly) the remaining content of files, which were not be stored in BS. At the delivery phases, users will send requests and information about stored data to the data center. Therefore, the transmission load will be the remaining ones (have not been stored) multiplying with K when the requests of uses are different. It is easy to find that the result of (12) is better than that of (8) as:

$$\left(1-\frac{M_u}{N}\right)\left(1-\frac{M_b}{N}\right)=1-\frac{M_u+M_b}{N}+\frac{M_u.M_b}{N^2} > 1-\frac{M_u+M_b}{N} \qquad (13)$$

Proposition 2:

For a communication system with N, K, Mb, Mu when the prefetched content is cooperated between Mb and Mu, The transmission load in backhaul line for coded caching is:

$$Q_{cod.BHj2} = \left(1-\frac{M_b}{N}\right)\frac{Q(K-m)}{m+1} \qquad (14)$$

Proof:

Files in data center are fn (n=1, 2,…, N), which are divided into C_K^j equal parts as in [3].

In caching phase: These parts are indexed and prefetched into users like as section 2.B.

But for the purpose of cooperation and for simplicity, we can denote these parts as f_n^j(j=1, 2,.. c_K^m). Fig. 8 illustrates the BS and user cooperation for the coded caching case.

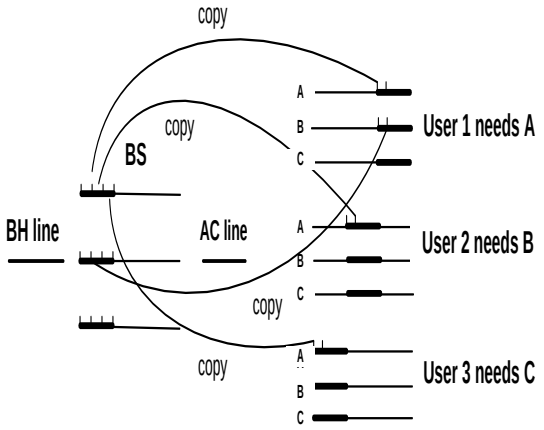

Fig. 8. The prefetched Mb và Mu cooperation.

When Mb=0, the transmission load from data center to users is as (6). BS does not play any role

When Mb>0, we copy the first part all of f_n^j that is Δf_n^j with the size of (Mb/N)(Q/C_K^m) to form a size of Mb/N for a file as prefetching in the memory of BS. This satisfies the memory condition of Mb.

With the above caching cooperation, the delivery phase is performed as the following.

When users have requests, if part f_n^j is in the appropriate multicast message, the data center only needs to send multicast message containing the remaining parts (f_n^j - Δf_n^j) to the BS. At the BS, it will automatically add Δf_n^j to form the complete multicast message for sending to user. It is clear that the transmission load on access line in this case is the same as (5), but on backhaul line the size of each equal:

$$f_n^j - \Delta f_n^j = \frac{Q}{c_K^m} - \frac{M_b}{N}\frac{Q}{c_K^m} = \frac{Q}{c_K^m}\left(1 - \frac{M_b}{N}\right) \quad (15)$$

Since we have c_K^{m+1} multicast message, the transmission load on backhaul line will be:

$$Q_{cod.BHj2} = \frac{C_K^{m+1}Q}{c_K^m}\left(1 - \frac{M_b}{N}\right) = \left(1 - \frac{M_b}{N}\right)\frac{Q(K-m)}{m+1} \quad (16)$$

It is clear that (16) gives better result than (11) as:

$$1 - \frac{M_b}{N} < 1 - \left(\frac{M_b}{N}\right)^{m+1} \quad (17)$$

Here, m >0 and Mb<N

Proposition 3:

Inspired from proposition 1, the system has Mb at BS equals to the system without BS but having the cache memory at users with the size of Mb+Mu.

For a system with N, K, Mb, Mu ((Mb/N+Mu)<N), the coded caching can be cooperated between BS and users, to the backhaul transmission load is:

$$Q_{cod.BHj3} = \left(1 - \frac{M_b}{N}\right)\frac{Q\left(K - m^{'}\right)}{m^{'} + 1} \quad (18)$$

With $m^{'} = \frac{K\left(M_b/N + M_u\right)}{N}$

Proof:

Following the proposition 2, we divide every file into $c_K^{m^{'}}$ parts.

$$m^{'} = \frac{K\left(M_b/N + M_u\right)}{N} \quad (19)$$

denoted each part as $f_n^{j'}$ (j'=1,2,.., $c_K^{m^{'}}$) with size of $Q/c_K^{m^{'}}$.If we prefetche these parts for each user following the rule in the section 2.B, equation (4) , the size of caching will be Q(Mb/N+Mu).

It will exceed memory of QMu at every user

In fact, we will only load the part of ($f_n^{j'} - \Delta f_n^{j'}$) for an user. At the same time the missing part is $\Delta f_n^{j'}$ with the size of (Mb/N) ($Q/C_K^{m^{'}}$) will be loaded at BS. It guarantees that both of memory size at BS and user is Mb and Mu.

Delivery phase: After receiving request from user, BS has two tasks: first, it sends the missing part $\Delta f_n^{j'}$ associated to the requested file to user following the unicast message to fill out corresponding parts of ($f_n^{j'} - \Delta f_n^{j'}$). When data center sends multicast message with the size of ($f_n^{j'} - \Delta f_n^{j'}$) like as Proposition 2, the BS adds the missing part $\Delta f_n^{j'}$ to form the complete multicast message and continusly send to users.

It is clear that the result in (17) has the transmission load is smaller than (15) when m'>m. However, it will need to pay for increasing of transmission load in access line since the unicast message has been used to fill out the missing part. This scenario can be accepted when the backhaul line is connected to many BS and it is needed to reduce the transmission load on backhaul.

4. Numerical results

This session evaluates the performance of the proposed joint BS and user caching cooperation. Numerical results and comparison between the proposed and the reference [2] are considered.

Fig. 9 shows the numerical results for transmission load with various cache sizes of M and for uncoded caching case (equation (8) and (12)). We assume that N=K=30, while M=Mb+Mu changing from 0:30 with different ratios: Mb/M=0; 1/2; 1/4.

From results obtained in Fig. 9, it can be concluded that:

Transmission load according to (8) is always greater under (12). This is more evident when M is large. (M=20--30).

Transmission load according to (8) is maximum when Mb/M=1/2.

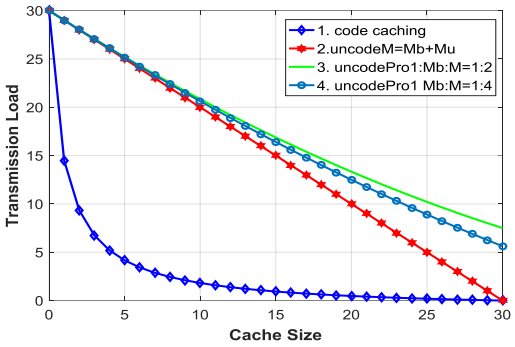

Fig. 9. Transmission load vs. Cache size for uncoded caching.

In the joint caching solution, two scenarios are examined:

Scenario 1:

The size of Mu changes from 0:30 while Mb = 15 is added. Experimental results in Fig. 10 show that by adding Mb, from (14), the result is better than Mb=0. However, comparing to pro.2 (14), results of pro.3 (16) are better. This is evident when M is small (M=2→8).

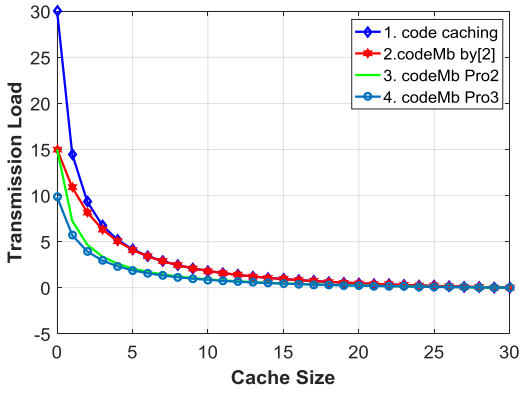

Fig. 10. Joint coded caching, scenario 1.

Scenario 2:

The sum of memory M=Mb+Mu changes from 0:30 in which Mb takes a part and follows the ratio Mb/M =0; 1/2.

Results in Fig. 11 show that when Mb=0, the memory is completely given for Mu, resulting in the highest performance when coded caching as (6).

When gives a part of total memory for Mb, results from proposition 2, 3 is inferior but still better from (11).

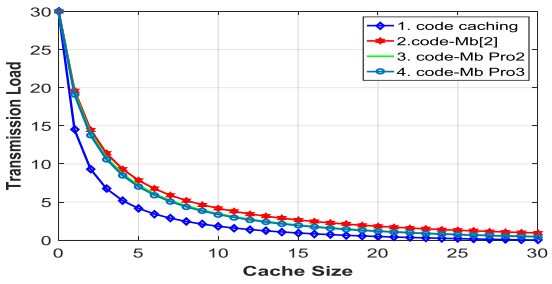

Fig. 11. Joint coded caching, Scenario 2.

5. Conclusions

This paper proposes a novel coded caching solution for the hierarchical communication system with a server to be connected with users through a BS. In this solution, the BS and users is co-opperated in both the pefetch and delivery phases. We have demonstrated that the proposed solution further improves the transmission load on backhaul line compared with the reference. Especially, if the transmission load on access line is added with with some unicast message, the overall load on the backhaul line can be more reduced. Several caching scenarios were also examined to demonstrate the efficiency of the proposed solution. In future work, we can extend the proposed method for a system with multiple BSs, which is widely used in real applications.

Acknowledgements

This work has been supported by Vietnam National University, Hanoi (VNU) under Project No.QG.18.39

The authors also thank the precious comments of Prof Nguyen Viet Kinh

References

[1] D. Liu, B. Chen, C. Yang, A.F. Molisch, Caching at the Wireless Edge: Design Aspects, Challenges, and Future Directions, IEEE Communications Magazine. 54 (2016) 22-28. https://doi.org/10.1109/MCOM.2016.7565183.

[2] T.X. Vu, S. Chatzinotas, B. Ottersten, Edge-Caching Wireless Networks: Performance Analysis and Optimization, IEEE Transactions on

Wireless Communications. 17 (2018) 2827-2839. https://doi.org/10.1109/TWC.2018.2803816.

[3] M.A. Maddah-Ali, U. Niesen, Fundamental Limits of Caching, IEEE Transactions on Information Theory. 60 (2014) 2856-2867. https://doi.org/10.1109/TIT.2014.2306938.

[4] M.A. Maddah-Ali, U. Niesen, Decentralized Coded Caching Attains Order-Optimal Memory-Rate Tradeoff, IEEE/ACM Transactions on Networking. 23 (2015) 1029 - 1040. https://doi.org/10.1109/TNET.2014.2317316.

[5] U. Niesen, M.A. Maddah-Ali, Coded Caching with Nonuniform Demands, IEEE Transactions on Information Theory. 63 (2017) 1146 - 1158. https://doi.org/10.1109/TIT.2016.2639522.

[6] Q. Yu, M.A. Maddah-Ali, A.S. Avestimehr, The exact rate-memory tradeoff for caching with uncoded prefetching, IEEE Transactions on Information Theory. 64 (2018) 1281 - 1296. https://doi.org/10.1109/TIT.2017.2785237.

[7] S.P. Shariatpanahi, H. Shah-Mansouri, B.H. Khalaj, Caching gain in interference-limited wireless networks, IET Communications. 9 (2015) 1269 - 1277. https://doi.org/10.1049/iet-com.2014.0955.

[8] N. Naderializadeh, M.A. Maddah-Ali, A.S. Avestimehr, Fundamental limits of cache-aided interference management, IEEE Transactions on Information Theory. 63 (2017) 3092-3107. https://doi.org/10.1109/TIT.2017.2669942.

[9] J. Hachem, U. Niesen, S. Diggavi, Energy-Efficiency Gains of Caching for Interference Channels, IEEE Communications Letters. 22 (2018) 1434-1437. https://doi.org/10.1109/LCOMM.2018.2822694.

[10] M.A. Maddah-Ali, U. Niesen, Cache-aided interference channels, IEEE International Symposium on Information Theory ISIT. (2015) 809-813. https://doi.org/10.1109/ISIT.2015.7282567.

[11] T.X. Vu, S. Chatzinotas, B. Ottersten, T.Q. Duong, Energy minimization for cache-assisted content delivery networks with wireless backhaul, IEEE Wireless Communications Letters. 7 (2018) 332-335. https://doi.org/10.1109/LWC.2017.2776924.

[12] S. Li, Q. Yu, M.A. Maddah-Ali, A.S. Avestimehr, Coded distributed computing: Fundamental limits and practical challenges, 50th Asilomar Conference on Signals, Systems and Computers. (2016) 509-513. https://doi.org/ 10.1109/ACSSC.2016.7869092.

[13] S. Li, M.A. Maddah-Ali, Q. Yu, A.S. Avestimehr, A fundamental tradeoff between computation and communication in distributed computing, IEEE Transactions on Information Theory. 64 (2018) 109-128. https://doi.org/10.1109/TIT.2017.2756959.

[14] S. Borst, V. Gupta, A. Walid, Distributed caching algorithms for content distribution networks, Proceedings IEEE INFOCOM. (2010) 1-9. https://doi.org/10.1109/INFCOM.2010.5461964.

[15] N. Karamchandani, U. Niesen, M.A. Maddah-Ali, SN Diggavi, Hierarchical coded caching, IEEE Transactions on Information Theory. 62 (2016) 3212-3229. https://doi.org/10.1109/TIT.2016.2557804.

[16] S.P. Shariatpanahi, G. Caire, B. H. Khalaj, Multi-antenna coded caching, IEEE International Symposium on Information Theory ISIT (2017) 2113-2117. https://doi.org/10.1109/ISIT.2017.8006902.

[17] R. Pedarsani, M.A. Maddah-Ali, U. Niesen, Online coded caching, IEEE/ACM Transactions on Networking. 24 (2016) 836-845. https://doi.org/10.1109/TNET.2015.2394482.