Cooperation Based Probabilistic Caching Strategy in Clustered Cellular Networks

Yifan Zhou, Zhifeng Zhao, Rongpeng Li, Honggang Zhang, and Yves Louet

***Abstract*—This paper will discuss the probabilistic caching strategies in spatially clustered cellular networks. Thanks to the content preference of mobile users, proactive caching can be adopted as a promising technic to diminish the backhaul traffic and to decrease the content delivery latency. However, basically there are two obstacles to accomplish the caching policy, i.e., the limited storage capacity of small cells to cache large amount of multimedia contents, and the too small number of users under each base station to imply the content aggregation effect. Traditional caching strategies of the base station only concern its local requests from the connected users through wireless links, but neglects the potential benefit from the cluster feature of the network infrastructure and user traffic demand. In this letter, we proposed a new policy called 'Caching as a Cluster', where small cells can exchange content with each other to fulfill every user request within the cluster of base stations. Intuitively, this cooperation between base stations makes a difference to decrease the content delivery latency of mobile users in clustered cellular networks as testified in our numerical simulation.**

***Index Terms*—Clustered cellular networks, probabilistic caching, content delivery latency.**

I. Introduction

In recent years, proactive caching has been widely investigated in cellular networks [1], which is motivated by the content preference of mobile user requests, to increases the network capacity and reduces the delivery latency. Mostly, researchers consider the BSs (Base Stations) as storage anchors [2] or even the UEs (user equipments) in D2D (device to device) scenarios [3]. However, there are two intrinsic obstacles to accomplish the caching policy in the wireless part of cellular networks, i.e., the limited storage capacity of single BS to cache large amount of multimedia contents, and the too small number of users under each BS to imply the content aggregation effect [4]. To solve those problems, we propose a probabilistic caching scheme in spatially clustered cellular networks, where all SBSs (small BSs) within the cluster can exchange stored contents cooperatively to fulfill different content requests which can be aggregated together to provide more apparent content preference phenomenon.

Recently, many researches focus on the small cell caching, such as the FemtoCaching proposed in [5], where the authors consider that UEs can access to several different SBSs. Similarly in [6], it's declared that the most popular content caching policy is not the optimal under the high coverage regime. In general, all these related works assume that UEs directly connect to the SBS which stores the demanding content [7], [8]. However, this kind of procedure actually increases the interference in wireless environment, thus decreasing the overall throughput of the whole cellular networks.

Recently, there is a tendency to centralize the base band units of nearby SBSs with fiber-based link, plenty of wired bandwidth would be available for the caching content sharing among the cluster of SBSs [9]. From this point of view, we can reformulate the content placing problem in wireless caching scenario adding the content sharing cooperation between SBSs, under which the interference can be mitigated [4].

This letter distinguishes itself in two dimensions. Firstly, the probabilistic caching strategy is considered rather than the deterministic placement approach, which usually makes the problem NP-hard thus no efficient solution. Secondly, unlike the BS separately serving paradigm, we proposed the cooperative caching in cluster distributed SBS scenario, which is more practical in real deployment.

Utilizing bandwidth between nearby SBSs, we show that the proposed caching policy outperforms the greedy and uniform caching strategy in different scenarios, in terms of the average delivery latency which takes the queuing latency into consideration. The paper is organized as follows, after introducing the system model and corresponding problem formulation in Section II, the solution is proposed in Section III along with the numerical results presented in Section IV. Afterwards, the conclusion is given in Section V.

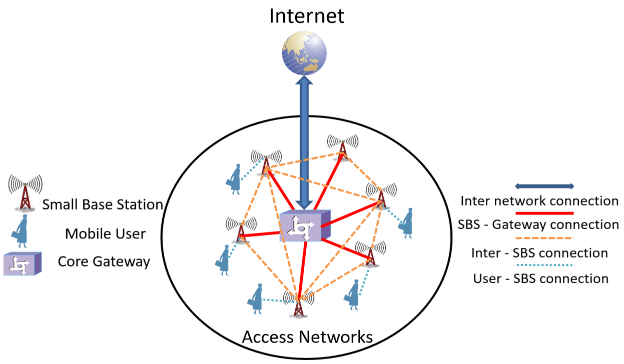

Fig. 1. Cooperative content caching in clustered cellular networks.

Y. Zhou, Z. Zhao, R. Li and H. Zhang are with the College of Information Science and Electronic Engineering, Zhejiang University, Hangzhou 310027, China (email:{zhouyftt, zhaozf, lirongpeng, honggangzhang}@zju.edu.cn). Y.Zhou and Y. Louet are with CentraleSupélec, Cesson-Sévigné 35576, France (email: Yves.Louet@centralesupelec.fr).

This work was supported in part by the National Postdoctoral Program for Innovative Talents of China (No. BX201600133), the Program Zhejiang Leading Team of Science and Technology Innovation (No. 2013TD20) and in part by the Project funded by China Postdoctoral Science Foundation (No. 2017M610369) and French National Research Agency (ANR) under No. ANR-15-CE25-0005-01.

II. Model Description and Problem Formulation

A. Model Description

Here, we consider a spatially clustered cellular network, which consists of a cluster of SBSs and their connected core gateway, as illustrated in Fig. 1. We assume that the MBS takes responsibility of the control plane, which orchestrates the placement and offloading of cached contents, thus not displayed in this data plane plot.

Utilizing the content preference of mobile users, SBSs can cache popular contents in their local storage to reduce the overall latency. Different from related works, here we assume that within the SBS cluster, cached contents can be exchanged between all SBSs to serve each user request. In detail, UE sends data requests to its connected SBS, then the SBS will firstly check whether the requested content is in its storage (local hit) or not. If not, it will offload this request to another available SBS according to a local reference table which is managed by the MBS, then this requested content can be transferred to UE through the inter-link between these two SBSs (cluster hit). Otherwise, the connected SBS should turn to the gateway which is assumed to contain the whole library (global hit). Generally, the delivery latencies are distinctive for different hit scenarios and routing paths, which is the basic principle here to design the caching strategy.

In detail, we assume that the storage capacity of each SBS is C_s and there are N_s equivalent SBSs within each cluster, where the bandwidth between each pair of SBSs is B. Additionally, the content requests rate under each SBS is λ, and the serving rate of each SBS is μ, with $\lambda < \mu$. Because of the requests offloading, the arriving rate for each SBS depends on the caching strategy, thus we adopt the queuing-based average waiting time for local and cluster hits, instead of constant latency. Furthermore, the global hit latency D_{gh} is assumed to be constant and significantly larger than those of local and cluster hits (D_{lh} and D_{ch}), due to the long transmission distance and larger serving capacity [10]. Above all, the imbalanced distribution of requested contents is represented as Zipf's distribution (assume each content has the same size $Z = 50MB$), then the probability of the ith most popular content is calculated as:

$$f_i = \frac{i^{-\gamma}}{\sum_{j=1}^{N_c} j^{-\gamma}}, \tag{1}$$

where N_c is the number of different contents, and γ is the skewness parameter. Related parameters and their typical values are depicted in Table I.

Therefore, the probabilistic distributed caching problem can be simplified to one question: given the limited storage capacity and offload bandwidth, how to cache the contents across all SBSs in order to minimize the average content delivery latency?

Actually, the caching probability of the ith content can be represented as $P_i \in [0, 1]$, which means that P_i proportion of all SBSs within the cluster has cached the ith content. Then, for a cached ith content request, the local hit probability will be P_i, and the cluster hit probability P_{ic} would be $1 - P_i$ and the global hit probability will be 0. Otherwise, if the content is cached nowhere within the cluster ($P_i = 0$, $P_{ic} = 0$), then there should be a global hit from the gateway, and its probability $P_{ig} = 1$.

TABLE I
Probabilistic caching parameters description.

Parameter	Description	Typical Value
λ	Requests rate under each SBS	100 s^{-1}
μ	Serving rate of each SBS	250 s^{-1}
N_s	Number of SBS in the cluster	10
C_s	Storage capacity of SBS (contents count)	20
B	Bandwidth between each pair of SBSs	800 Mbps
D_{gh}	Delivery latency from core gateway	30 ms
N_c	Number of different contents	100
Z	Size of each content	50 MB
γ	Skewness of Zipf's law	0.65

Besides, the corresponding latency for three different kinds of cache hit can be characterized as follows. The local hit experiences a single queue during the request, and the average processing time is $1/(\mu - \lambda_a)$, where λ_a is the sum of the original user requests rate λ and the offloaded requests rate λ_o. Specifically, the offloaded rate consists of all cluster hit requests and are evenly orchestrated to all the SBSs within the cluster, as derived in the following equation.

$$\lambda_a = \lambda + \lambda \sum_{j=1}^{N_c} P_{ic} f_i. \tag{2}$$

Furthermore, as stated in the delivery process, the cluster hit experiences two equivalent queues during the request, where one is in the connected SBS and the other one is in the offloaded SBS. Therefore, the average latency for the cluster hit will be double of the local hit, with quantity $2/(\mu - \lambda_a)$. Above all, the latency for the global hit is assumed to be constant.

B. Problem Formulation

As the request of each content experiences a specific average latency according to its caching strategy, i.e., the P_i. Therefore, we can derive the overall average latency by multiplying the requested probability of each content by its correspondent average latency. Furthermore, the average delivery latency of a specific content is related to its hit pattern, together with the corresponding latency, as detailed in the next paragraph. The objective here is to minimize the average content delivery latency of a random user, which can be derived from the average latency of all contents in the probabilistic caching scenario. Therefore, the probabilistic caching strategy is reduced to a latency minimization problem with the variables P_i and the limited storage and bandwidth constraints as follows:

Minimize:

$$\sum_{i=1}^{N_c} f_i (P_i D_{lh} + P_{ic} D_{ch} + P_{ig} D_{gh}) \tag{3}$$

Subject to:

$$\sum_{i=1}^{N_c} P_i \le C_s, \tag{4}$$

$$\lambda Z N_s \sum_{i=1}^{N_c} P_{ic} f_i \leq B \binom{N_s}{2}, \tag{5}$$

where (3) represents the average latency in the contents' point of view, and f_i is the requested probability of the ith most popular content. For the constraints, (4) implies the storage limitation of SBSs and (5) means that the amount of all transferred contents within the cluster (left part) cannot exceed the overall bandwidth between all SBSs (right part) which is a combinatorial number. Here, we assume that the transferred traffic is evenly offloaded on connections between each pair of SBSs.

III. Proposed Solution

Traditionally, SBSs are preferred to cache popular contents in order to reduce the overall latency. However, in addition to the storage and bandwidth constraints, the queuing latency introduced in our model makes the problem more practical but even more complicated. To solve the problem, we introduce the intermediate parameter S, which is the number of contents that are distributively cached in the cluster of SBSs. Hence, there are $N_c - S$ contents that should be delivered by global hit from the gateway which experiences a higher latency. Combining this definition and the latency description in Section II, we can reformulate the problem into:

Minimize:

$$\sum_{i=1}^{S} f_i \frac{P_i + 2P_{ic}}{\mu - \lambda(1 + \sum_{i=1}^{S} f_i P_{ic})} + \sum_{i=S+1}^{N_c} f_i D_{gh} \tag{6}$$

Subject to:

$$\sum_{i=1}^{S} P_i \leq C_s, \tag{7}$$

$$0 \leq P_i \leq 1, \tag{8}$$

$$\sum_{i=1}^{S} (1 - P_i) f_i \leq B_0 = B(N_s - 1)/(2\lambda Z). \tag{9}$$

Given S, it can be shown that the average latency in (6) is increasing with the left part of (9), which is the portion of requests that need to be offloaded. Therefore, to achieve the minimal latency, there are two steps to proceed: first is to maximize the intermediate variables S while fulfilling all these constraints, since the more contents are cached in the SBSs, the smaller is the average latency to fetch a random content. Then, for each S, proper values (integral multiple of $1/N_s$) are assigned to P_i of all S contents in order to reach the minimum of (6).

Actually, since the sequence $\{f_i\}$ is decreasing with i, the minimal value of the left side of (9) (denoted as Q) is 0 when $S \leq C_s$, but it increases with S when $S \geq C_s$. Therefore, to make S maximal, we increase S gradually to assure that the minimal value of Q cannot exceed B_0. As the S increases to $S + 1$, a portion of $1/N_s$ should be shifted from another P_i to P_{S+1}, to make sure that $1/N_s \leq P_i$ stands for each i smaller than S. After Algorithm 1 is processed, the caching proportion of each content is derived, based on which we can calculate the average content delivery latency with (6).

Algorithm 1 Probabilistic Strategy for Cluster Caching.

```
Input: $\{f_i\}$, $C_s$, $B_0$, $N_s$, $N_c$;
Output: $\{P_i\}$, $S$;
Initialize: $\{P_i\} = 0$, $Q = 0$, $j = 1$;
while ($j < N_s$) do
    $P_j = 1$;
    $j = j + 1$;
end while
while ($Q \leq B_0 \bigwedge j < N_c$) do
    $P_j = \frac{1}{N_s}$;
    $P_{C_s - \lfloor \frac{j-1}{N_s-1} \rfloor + 1} - = \frac{1}{N_s}$;
    $Q = Q + \frac{1}{N_s}(f_{C_s - \lfloor \frac{j-1}{N_s-1} \rfloor + 1} - f_s) + f_s$;
    $j = j + 1$;
end while
$S = j - 1$;
return $\{P_i\}$, $S$;
```

IV. Numerical Results

In this section, we present the numerical results of our proposed algorithm compared with the uniform caching algorithm (contents are equally cached under bandwidth constraint) and the greedy algorithm (always cache the most popular contents). In detail, we depict the average latency performance according to the typical parameter setting as presented in Table I.

As seen in Fig. 2, the average latency increases with the request rate for all three strategies. On one hand, it's caused by the increasing of queuing latency in each SBS, while on the other hand, it's due to the relative decrease of available bandwidth. Besides, the latency increasing of greedy strategy with respect to request rate is slower than that of uniform and proposed strategy, because of the increasing effect of offloaded traffic on the queuing latency. When the request rate is relatively low, the uniform strategy outperforms the proposed strategy since it can cache more contents within the cluster, while does not increase queuing latency much.

Besides, we investigate the effect of bandwidth between SBSs on the average latency, as depicted in Fig. 3. Clearly, the average delivery latency of our proposed algorithm is significantly decreasing with the bandwidth, as more and more contents can be shared between different SBSs, shorting the delivery path of user requests. However, the greedy algorithm doesn't make use of the bandwidth advantage thus shows a flat line.

Besides, all three strategies benefit from the increasing capacity of SBS, while our proposed policy outperforms the greedy one and the uniform solution as depicted in Fig. 4. The uniform strategy is approximate with the greedy one, which means that the available bandwidth is inadequate to store more contents than greedy strategy.

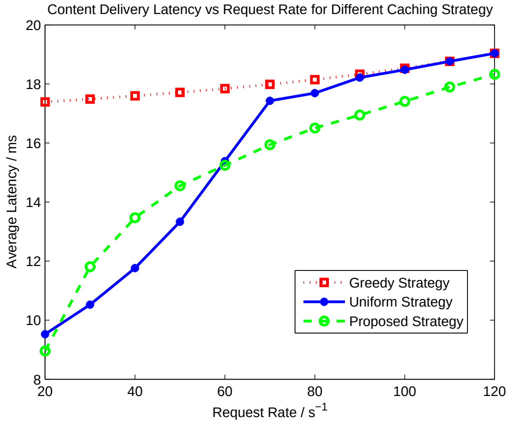

Fig. 2. Average latency with respect to different request rates of each SBS.

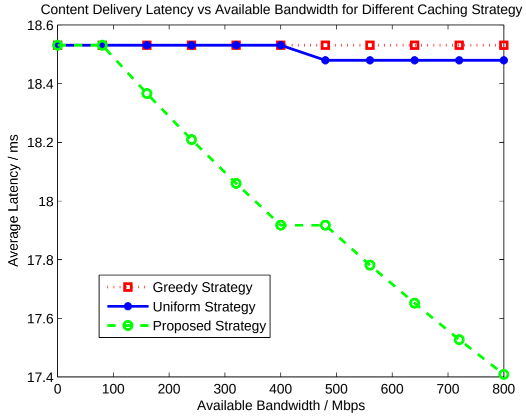

Fig. 3. Average latency with different bandwidths between each SBS pairs.

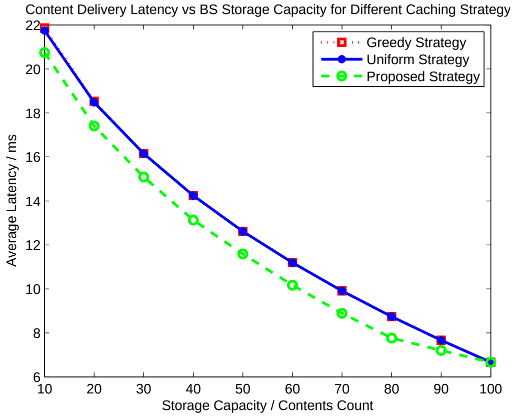

Fig. 4. Average latency with different storage capacity of SBS.

After all, we investigate the effect of Zipf's skewness on the average latency performance, as shown in Fig. 5. Clearly, it shows that the average latencies decrease as the skewness increases for all three polices, which implies contents popularity are more unevenly distributed. Comparatively, our proposed algorithm always outperforms the other two with a more than 5% less latency.

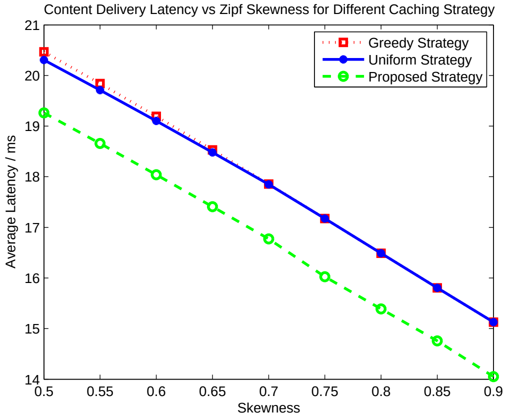

Fig. 5. Average latency with different skewness of the Zipf's distribution.

V. Conclusion

In this letter, we investigate the probabilistic caching problem in a spatially clustered cellular networks scenario where the interchange bandwidth between SBSs can be utilized for content sharing. To reduce the average content delivery latency which takes the queuing latency into consideration, we proposed a more efficient probabilistic solution for the caching placement problem compared to the traditional greedy policy and uniform strategy. In our proposal, contents with different popularity is distributed across the SBS cluster with different caching probability, which is derived from a average latency minimization algorithm. According to the numerical results, our solution achieves a much better and more flexible performance than traditional strategies.

References

[1] E. Bastug, M. Bennis, and M. Debbah, "Living on the edge: The role of proactive caching in 5g wireless networks," *IEEE Communications Magazine*, vol. 52, no. 8, pp. 82–89, 2014.

[2] D. Liu, B. Chen, C. Yang, and A. F. Molisch, "Caching at the wireless edge: Design aspects, challenges, and future directions," *IEEE Communications Magazine*, vol. 54, no. 9, pp. 22–28, 2016.

[3] M. Ji, G. Caire, and A. F. Molisch, "Wireless device-to-device caching networks: Basic principles and system performance," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 1, pp. 176–189, 2016.

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

[5] 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*, vol. 59, no. 12, pp. 8402–8413, 2013.

[6] B. Blaszczyszyn and A. Giovanidis, "Optimal geographic caching in cellular networks," in *Proceedings of ICC, IEEE*, 2015, pp. 3358–3363.

[7] K. Li, C. Yang, Z. Chen, and M. Tao, "Optimization and analysis of probabilistic caching in n-tier heterogeneous networks," *arXiv preprint:1612.04030*, 2016.

[8] Y. Chen, M. Ding, J. Li, Z. Lin, G. Mao, and L. Hanzo, "Probabilistic small-cell caching: Performance analysis and optimization," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 5, pp. 4341–4354, 2017.

[9] I. Chih-Lin, J. Huang, R. Duan, C. Cui, J. X. Jiang, and L. Li, "Recent progress on c-ran centralization and cloudification," *IEEE Access*, vol. 2, pp. 1030–1039, 2014.

[10] Q. Xu, J. Huang, Z. Wang, F. Qian, A. Gerber, and Z. M. Mao, "Cellular data network infrastructure characterization and implication on mobile content placement," in *Proceedings of the SIGMETRICS, ACM*, 2011, pp. 317–328.