Distributed Caching in Wireless Cellular Networks Incorporating Parallel Processing

Yue Sun*, Ziming Zhu, *Member, IEEE*, and Zhong Fan*

Toshiba Research Europe Ltd. Telecommunications Research Laboratory, Bristol, BS1 4ND, UK

Abstract

Distributed caching is a promising technique for reducing the redundant data traffic and user's content access delay in the telecommunication system. This article explores caching technologies with a focus on the processing of content requests in today's hierarchical wireless cellular networks. We observed that, as the number of caches at different layers of the network increases, the disadvantage of the hierarchical architecture in terms of processing delay gradually emerge. We introduce a parallel processing strategy in order to improve the efficiency of cache servers. Theoretical analysis and numerical simulations show the potential of the proposed scheme in terms of reducing both the content access delay and redundant data traffic in the core network. We also carry out cost assessments for the proposed scheme. Future research on related technologies is discussed at the end of this article.

Index Terms

distributed caching; content access delay; parallel processing.

I. Introduction

The popularity of smartphones drives rapid growth of the demand in terms of Internet data, especially for popular social media content. The total smart phone subscriptions are expected to reach 6.1 billion in 2020 [1]. The future wireless cellular networks need to cope with such massive demand with guaranteed quality of service, such as ultra-low content access delay and super-fast download speed. Energy efficiency is also a main concern in the development of communication architectures and protocols. In today's Internet architecture, which is based on centralised servers in the core network (CN) delivering files to all end users, bottlenecks occur as the centralised servers could be severely overloaded with numerous content request processing, resulting in long delays as well as network congestion. When multiple users request the same content, the very same data packets will be transmitted every time, causing significant redundant data traffic in the core network. Such systems have very low efficiency.

Caching frequently requested content in the distributed intermediate nodes of the network is a promising method to resolve this issue. Facilitated by distributed caching, local area servers that are closer to the users are able to process the requests and provide the data service directly if the requested content is stored. In such a system, the performance of the core network can be improved as the workload is partly offloaded to the edge of the network while users benefit from fast access to popular content on the Internet. Edge servers such as wireless base stations and local area network gateways have become more capable in terms of computing and storage recently.

* This work was carried out while Yue Sun was an MSc student from the University of Bristol working as an intern at Toshiba Research Europe Ltd., and Zhong Fan was a Chief Research Fellow of TREL.

Distributed caching has attracted major attention in a wide range of research and development for communication systems. For example, in the 3rd Generation Partnership Project's Long Term Evolution Advanced (3GPP LTE-A), also considered as the 4th generation (4G) cellular network, distributed caching can be implemented at the evolved packet core (EPC), the radio access network (RAN) and even at user devices. EPC mainly refers to the serving gateways (S-GW), packet data network gateway (P-GW) and mobility management entity (MME) that are connected mutually between remote Internet servers and wireless base stations. Caching is also believed to be an important functionality provided in the upcoming 5th generation (5G) networks [2]. By implementing caching at wireless base stations (or eNodeBs), Wi-Fi access points and WiMAX access points in the RAN, data service can be provided without the need of the backhaul transmission. The performances of caching at various layers of the wireless network, e.g., at the small base stations and at the user terminals were studied in [3]. The work in [4] investigated the caching scheme in the RAN based on the concept of content-centric networking, in order to minimise the content access delay of all users as well as redundant traffic in the network. A proactive caching is proposed in [5] to leverage the existing heterogeneous cellular networks and design predictive radio resource management techniques to maximise the efficiency of the network. The work in [6] explored the emerging device to device (D2D) communication technology and its potential for caching, where user devices are able to exchange data directly with each other in order to further reduce the pressure on wireless resources of the RAN. Researchers have also been working on caching strategies including what content, as well as how the data can be cached at various caches. Techniques including cooperative and coded schemes under different pricing ~~scenario~~, types of data traffic, and network constraints are proposed [7]. The work in [8] investigated the caching system where the content requests have both elastic and inelastic delay requirements.

In this article, we explore caching ~~network~~ with the focus on the processing of users' content requests in the current multi-layer wireless cellular networks. We observed that, as the number of caches at different layers of the network increases, the disadvantage of the hierarchical architecture in terms of processing delay gradually emerge. ~~A cache~~ hit ratio (CHR) is used as a key performance metric for caching. It denotes the percentage of content requests that can be serviced by using locally cached data. According to [9], the CHR is usually very low and variable ~~and~~ in small population regions, while reasonably high and stable where the number of users is very large. Therefore, the cost-effectiveness can be low when considering the time and energy spent on request processing and local content searching. In the worst case, the users may suffer even longer delays with local caching. We introduce a parallel processing strategy in order to improve the efficiency for the caching networks. Theoretical analysis and numerical simulations show the potential of the proposed scheme in terms of reducing both the content access delay and redundant data traffic in the core network. We also carry out cost assessments for the proposed scheme.

The rest of this article is organised as follows. The content request processing strategies and the parallel processing method ~~is~~ described in Section II. In Section III, we formulate theoretical analysis of the performance metrics of the proposed technique considering both non-interactive and interactive caching scenarios. We further evaluate the technique in terms of signalling overheads in Section IV. The above theoretical analysis is illustrated in Matlab based simulations. Section V draws the conclusion and indicates future research topics.

II. Hierarchical Caching with Parallel Processing

Figure 1 illustrates a typical LTE-A cellular network architecture. The mobile user equipment (UE) connects to the evolved universal terrestrial radio access network (E-UTRAN) via the base station (BS), then the data traffic converges to the serving gateway (S-GW). The mobility management entity (MME) is responsible for idle mode UE paging and tagging procedure. After that, the data traffic is transferred to the packet data network gateway (P-GW) and routed to the Internet. The reader is

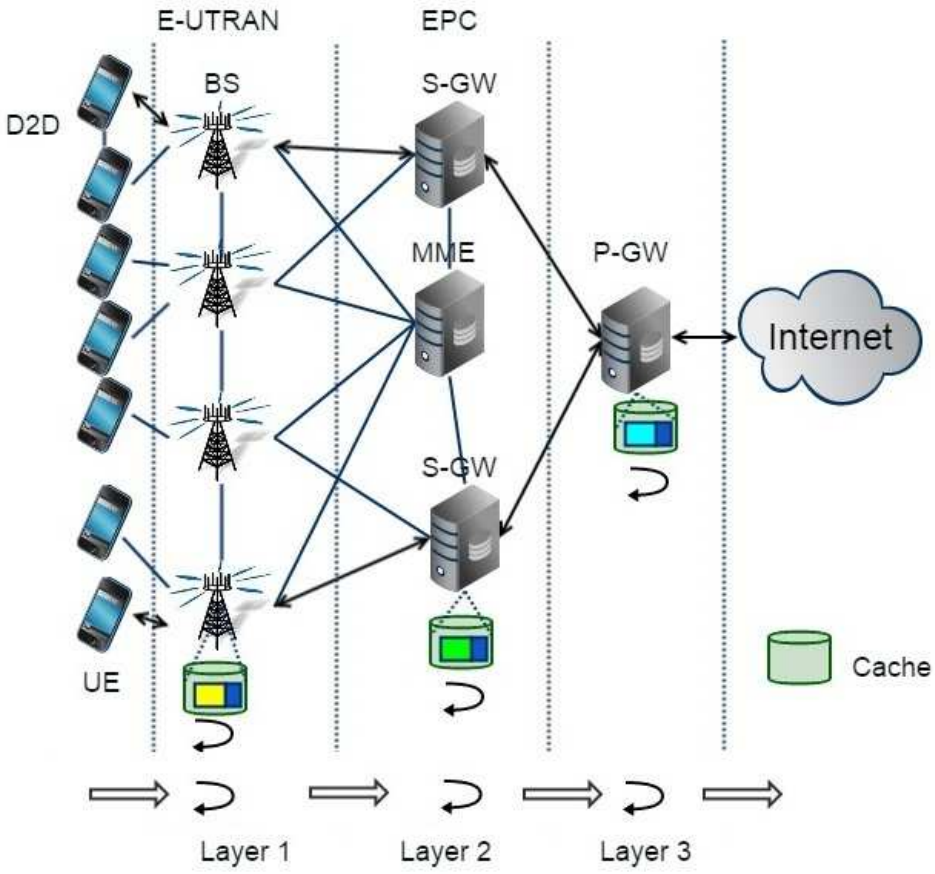

Fig. 1: Generalised 4G mobile cellular network architectures with caching.

referred to ~~The~~ 3GPP standards for more details [10].

The centralised data access architecture is very inefficient in coping with the exponentially growing traffic. The traffic aggregation points have to deal with enormous data request and duplicate traffic [4]. Adding caches in these points to store certain contents can reduce data access delay as well as duplicate transmission from the servers in the core network. In Figure 1, the solid lines denote the transmission links for data and control signal. The light green boxes denote the distributed caches. Various popular contents stored at the caches are represented by different colours. The caches of different layers ~~can form a~~ coordination where every cache stores different items in order to avoid redundant storage and improve caching efficiency. The black arrows beside the caches denote the content searching process. In existing caching networks, 'Suspend-and-Wait' method is often used for the processing of content requests in the cache [9], [11]. This means that the time delay for data access will increase when a request from the client passes through the cache. The processing of content requests is shown by the arrows at the bottom of Figure 1. The delay can become so large that outweighs the benefit of using caching as the number of caching layers increases. In particular, when the CHR is very low, such waiting time can be meaningless. To resolve this, we propose a parallel processing based caching strategy.

The main idea of the proposed technique is that, when a data request arrives at a node with a cache, the node immediately passes the request to the upper layer node without suspending and waiting. At the same time this node also searches for the requested information in its cache. The nodes at the upper layers carry on the same process until the request is passed to the server in the core network. This data requesting process can be seen as a combination of the original process without caching, and a cache searching in parallel. When a cache finds the requested information and makes sure it is valid, the cache

will send a 'negative request' to the upper layer nodes of the transmission path immediately. The requested information will be sent down to the client. Having received the negative request, the upper layer nodes can stop content searching (and data transmission) for the corresponding original request and start to deal with the next request in its queue. If a node does not find the requested information when it finishes searching its cache, then it can just discard this request. One possible situation is that after the core server (or an upper layer node) starts to send the requested data to the client, a node in the lower layer finds a valid copy in its cache. Then this node will send a negative request upwards. Having acknowledged the negative request, the upper layer network stops providing the data. The rest of the requested content will be sent from the cache at the lower layer.

Ideally, this method can almost eliminate the delay in searching for cached data, and significantly reduce the redundant traffic in the network. In particular when the cache's CHR is low, the proposed method has a great advantage. Although additional traffic overhead and power consumption for exchanging control message are required at the data aggregation nodes, the added value of the proposed caching strategy is still very obvious.

III. Non-interactive Caching Scenario

In this paper, we use content access delay and data traffic reduction as the main objectives to evaluate the performance and benefits of various caching strategies. Note that we only consider the reduction of the data traffic that is suitable for caching. Certain data traffic containing for example user privacy and security information is not included in this analysis. For simplicity, we assume that all the caches have already stored the popular files, such as the most recent news videos and images, according to a ~~prior~~ knowledge of the content requests. We also assume that the nodes with cache only communicate with other nodes in parent-child layers, not with their sibling nodes.

A common data transmission process can be divided into two relatively independent parts, namely the data request ~~sending~~ to the server and the requested data ~~sending~~ back to the client. For the second part, the delay means the first data packet that comes from the server (or cache) arrives at the corresponding client. The uplink and downlink transmission latency are also assumed to be equal. Now, we discuss the two parts respectively.

A. Sending content requests to the core network

Let $t_{i,j}$ denote the transmission time delay from layer i to layer j in a hierarchical network including routing and other processing time, and t_i denote the time consumed for local content searching in cache i. The caches at different layers of the communication path cannot interact with each other. This means that the caches do not have the information of what has been cached at where. They also do not exchange information to form any coordination to avoid redundant caching.

Denote t_{ul} as the uplink delay, which consists of the time of transmitting the content request through the network as well as the time the servers use for ~~request processing~~ and searching for requested information in the caches. When there is no cache in the network architecture, the uplink delay is only caused by the transmission. In this case, we write $t_{ul}^{nocache} = t_{1,N}$, where N is the total number of layers of the network.

When caches exist, a probability, namely CHR, is introduced as the requested content can be provided by certain cache in the network. In this case, the cache processing time is added to the intermediate layers. We use p_i to denote the CHR for layer i. If the content is not found at cache i, which has a probability of $(1 - p_i)$, the request will be sent further up and towards the core network. Therefore, the uplink delay is calculated according to the time spent in each layer of the network, and the

probability of that. We express that as

$$t_{ul}^{cache} = [t_1, (t_2 + t_{1,2}), (t_3 + t_{2,3}), ..., (t_N + t_{N-1,N})] \cdot \left[1, (1-p_1), (1-p_1)(1-p_2), ..., \prod_{i=1}^{N-1}(1-p_i)\right]^T \quad (1)$$

where $(\cdot)^T$ is used for vector transpose.

When the proposed parallel processing method is adopted, request transmission and content searching are processed in parallel, the delay is formulated as

$$\overline{\overline{t_{ul}^{cache}}} = [t_1, (t_2 + t_{1,2} - t_1), ..., (t_N + t_{N-1,N} - t_{N-1})] \cdot \left[1, (1-p_1), (1-p_1)(1-p_2), ..., \prod_{i=1}^{N-1}(1-p_i)\right]^T . \quad (2)$$

It is observed that an additional term $-t_i$ is introduced in the description of the delay between the adjacent layers. This is due to the content searching at layer i is processed in parallel while the request is being sent and processed at the upper layer. This overlapping time should be deducted. This is the main advantage of the parallel processing. In addition to the reduction in delays, the expected reduction of data traffic between different layers can be calculated. For example, in Figure 1, the reduction between the P-GW and the Internet can potentially reach $\prod_{i=1}^{3}(1-p_i)$ ~~percent~~ by using caching.

B. Sending the requested data to the client

Similar to the above analysis, we denote t_{dl} as the time a server spends for sending the requested content to the corresponding client. For simplicity, when there is no cache in the network, we consider $t_{dl}^{nocache} = t_{ul}^{nocache}$. When caching is deployed, data may be provided by intermediate caches, subject to certain CHR. Hence the delay is expressed as

$$t_{dl}^{cache} = [t_{N,N-1}, t_{N-1,N-2}, ..., t_{2,1}] \cdot \left[\prod_{i=1}^{N-1}(1-p_i), \prod_{i=1}^{N-2}(1-p_i), ..., (1-p_1)\right]^T , \quad (3)$$

As parallel processing only affects the uplink, the downlink delay remains the same as t_{dl}^{cache}. Therefore, the total data access delay in the caching network incorporating parallel processing is $\overline{\overline{t_{ul}^{cache}}} + t_{dl}^{cache}$.

C. Numerical Results

We simulate a simplified wireless cellular network in order to evaluate various caching strategies and show the performance of the proposed technique in terms of reducing access delay and redundant traffic. Transmission time between two levels is randomly assigned ranging from $15ms$ to $20ms$ and the time spent in each cache ranges from $2ms$ to $5ms$. We refer to the LTE-A networks when setting specific values for data transmission and network operation parameters [10]. Considering that such setting of parameters may have practical limitations, the simulation results reflect the performance of the proposed technique qualitatively.

To the best of our knowledge, there is no determined relationship between the CHR value and the location of the caches. Generally, the caches closer to the core network have higher CHRs. It is possible that lower layer caches achieve higher CHR. Therefore, in this study, the CHR values are generated randomly for each cache while the caches at higher layers has a higher averaged value. The system performance curves are shown in Figure 2, with the average CHR of all caches increases from 0 to 0.2. As shown in Figure 2 (a) and (b), when the CHR is very low, i.e., the requested data is not likely to be stored in the caches, the system without any caching (the pink lines) generates the lowest data access delay. As the CHR increases, the effect of caching in terms of delay reduction can be gradually seen, as more data can be provided by the distributed caches

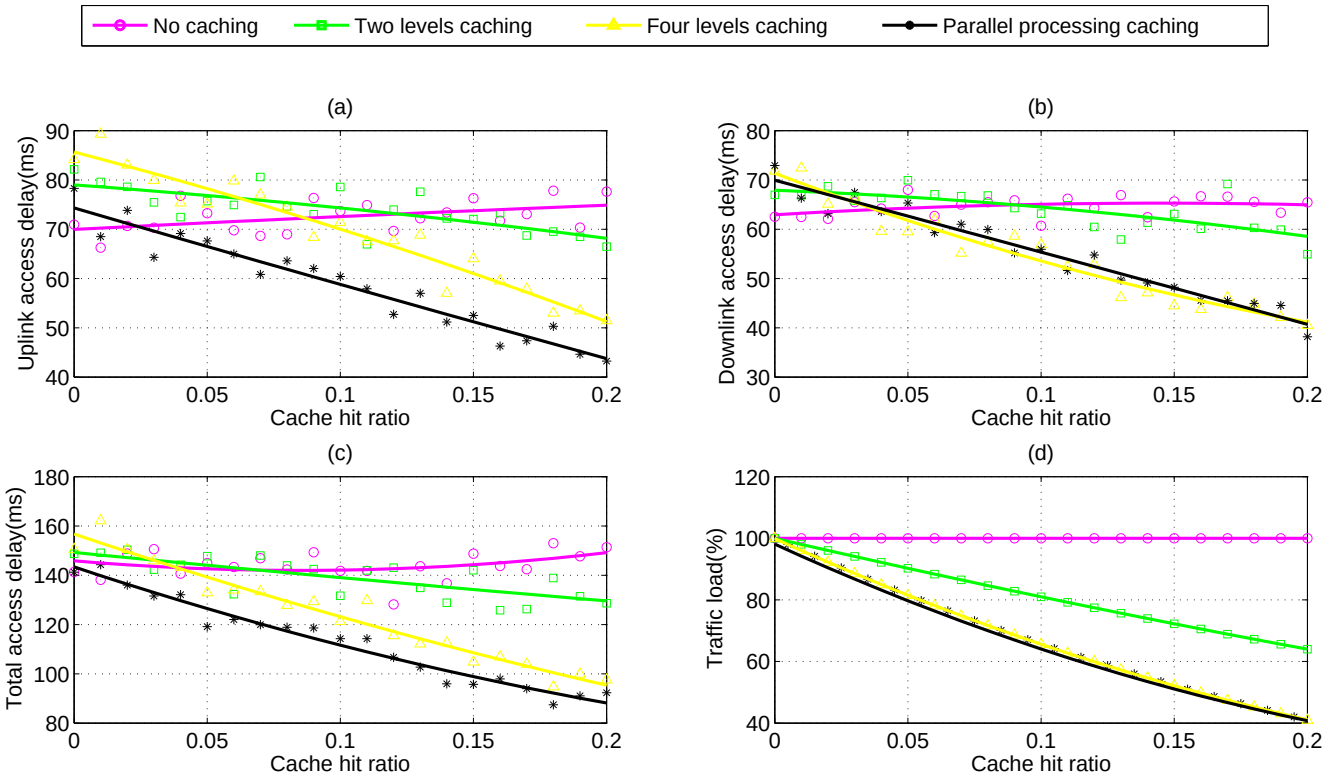

Fig. 2: Performance evaluation of hierarchical caching: (a) uplink access delay; (b) downlink access delay; (c) total access delay; (d) traffic load at P-GW level.

that are closer to the clients. When the caches have a relatively high CHR of 0.2, an approximately 30% delay reduction can be provided by using caching at all of the four layers of the network (the yellow lines). By adding the proposed parallel processing to the caching network (the black lines), the uplink delay can be further reduced by around $5ms$ (7%).

It can be seen in Figure 2 (c) that the parallel processing caching strategy has the best performance. When the CHRs are 0.05, 0.1 and 0.2, the content access delays reduce by 15%, 18%, and 40% ~~comparing~~ to no caching, respectively. It also produces an average of $10ms$ further delay reduction ~~comparing~~ to the four layers caching approach. Due to the use of parallel processing to save the time spent for content searching in the caches, it is also efficient even when the CHR of the caches is low. As the number of caches and CHR gradually increase, the traffic loaded at the P-GW level obviously decreases in Figure 2 (d). When the CHRs are 0.05, 0.1 and 0.2, the traffic loads can be approaching 78%, 63%, and 40% of original traffic load in the core network respectively. Note that the ~~track of~~ yellow lines overlap the black lines in Figure 2 (b) and (d). This is because since the proposed technique mainly focuses on reducing the uplink delay, the performance in terms of downlink delay and traffic reduction are likely to be the same with or without parallel processing. It is worth ~~to note~~ that each time the cache finds the requested information, the cache will send the negative request through the network. An additional signalling overhead is generated for the exchange of control message. However, considering that the amount of extra traffic is usually negligible compared with the requested data, we do not reveal it in Figure 2.

IV. Interactive Caching Scenario

~~An~~ interactive caching method ~~means~~ the caches in different layers can interact with each other. Caching devices can potentially form a coordination in order to reduce duplicate content cached in more than one layer. As a result, the system CHR is increased, and thus the redundant data traffic in the core network can be further reduced. Without detailed discussion

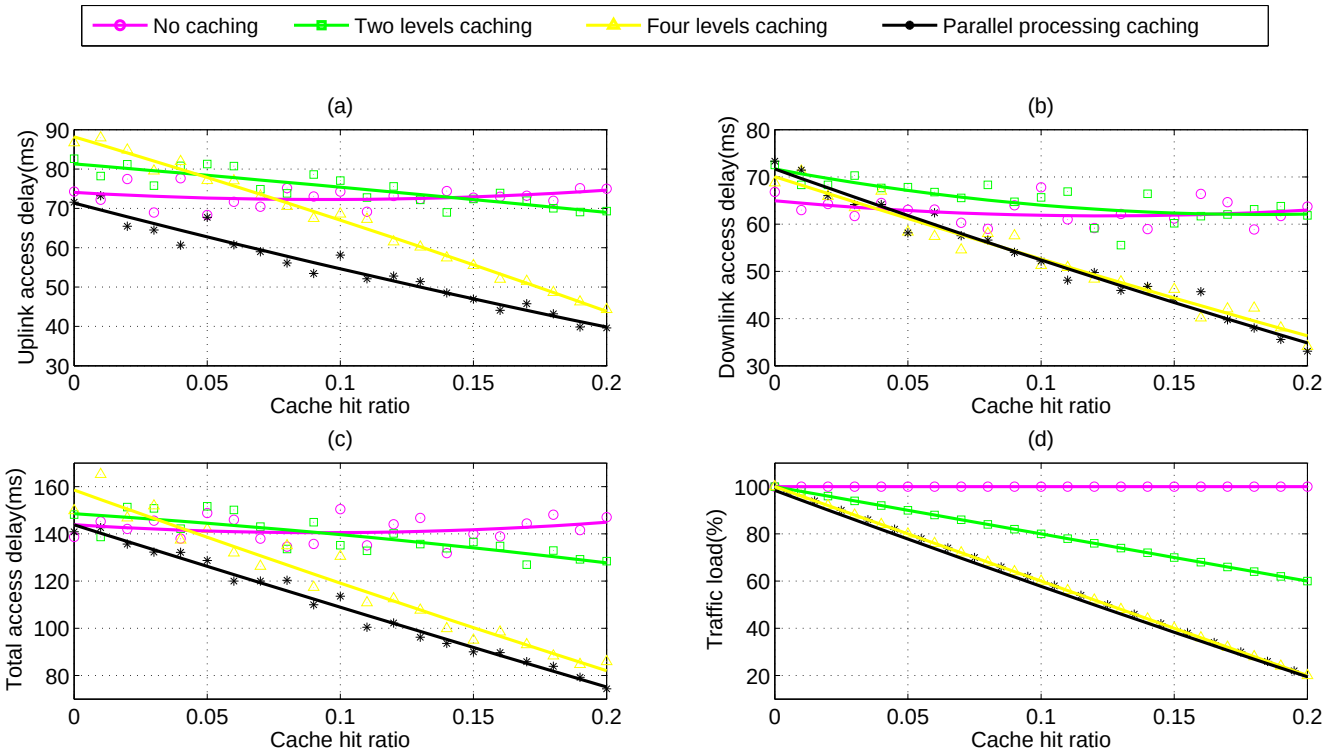

Fig. 3: Performance evaluation of hierarchical caching (interactive scenario): (a) uplink access delay; (b) downlink access delay; (c) total access delay; (d) traffic load at P-GW level.

of the enabling techniques, we assume an ideal scenario where the data files stored in every cache are totally different.

A. Delay Analysis

The method for the derivation of processing delay and traffic reduction ~~are~~ similar as before. However, we pay particular attention to the difference in the system CHR after considering the interactive performance of the caches. In particular, the probability of a request being processed at each layer is modified as

$$\tilde{p} = \left[1, (1-p_1), (1-p_1-p_2), ..., (1-\sum_{i=1}^{N-1} p_i)\right]^T . \quad (4)$$

Comparing with 1, it is seen that $1-\sum_{i=1}^{N-1} p_i \leq \prod_{i=1}^{N-1}(1-p_i), \forall 0 \leq p_i \leq 1$ holds. The uplink delay is hence smaller. Thus the total data access delay can be further reduced.

B. Numerical Results

It is seen in Figure 3 (c) that using the proposed technique, the content access delay can be reduced considerably by ~~nealy a half~~, as the CHR increases to 0.2. An approximately $10ms$ additional reduction is achieved ~~comparing~~ to the non-interative scenario as in Figure 2 (c). This is because ~~that~~ the system overall CHR is increased by interactive caching. In addition, compared with Figure 2 (d), the traffic load which is also visibly influenced by the CHR, declines significantly. The traffic loads can be approaching 20% of original traffic load in the core network when the CHR reaches 0.2. From the figures shown above, we can easily find that the parallel-processing method (the black lines) is obviously better than the non-parallel method under the same circumstances. As the CHR increases, the time delay and traffic load are both becoming smaller which are

beneficial for both the customers and the carriers. In general, interactive caching is able to achieve better performance than the non-interactive case. Various methods are proposed in [12], [13] to realise the cooperative function of content caching.

In the two sets of simulation results, we can see large variations in term of traffic loads. However, in practice, the performance of the traffic reduction by using caching is more complicated and depends on the demand of the clients, capabilities of individual servers, and more importantly the volatile operating conditions of wireless cellular networks. Therefore, such performance gains might not be seen in actual situations. However, we are optimistic that with the combination of various optimisation techniques and efficient network management tools, the proposed caching strategy is able to achieve an improved performance in wireless networks.

V. Cost analysis of caching networks

A. Calculation of costs

We now study the economic overhead of utilising caching networks incorporating the proposed parallel processing technique. As it is too complicated to capture the large number of cost parameters in practical networks, which also vary case by case, we introduce a simplified cost model to analyse the cost-benefits for utilising caching [9]. A set of parameters are selected ~~as~~ to best indicate the difference in operational costs with and without parallel processing. We analyse the cost of data transmission from sending the user's request until the corresponding data has been received by the user. Note that the costs for the installation and maintenance of the communication infrastructure are not considered, as such costs are independent of whether distributed caches are used or not. We further assume that at the RAN, a user device connects to the same base station from sending a content request until receiving the data completely, hence there will be no additional cost for handover.

Let v denote the average size (in bytes) of a requested content and $\dot{v}$ be the size of the request, which is usually much smaller than v. We use $l_{i,j}$ to denote the total number of content requests between the two layers. Define $B_{i,j}$ and $B'_{i,j}$ as the uplink and downlink transmission cost per byte on the network between the layers i and j, respectively. We use '$'$' to exclusively represent downlink. The traffic volume between i and j is $V_{i,j} = l_{i,j}\dot{v}$ and $V'_{i,j} = l_{i,j}v$. The cost for data transmission can be simply calculated as $C^{tr}_{1,N} = V_{1,N}B_{1,N} + V'_{1,N}B'_{1,N} + O$. Here, O denotes the sum of other expenses, such as renting the core network services. When caches (CHR> 0) are deployed, not all of the content requests are required to be sent to the core network in the uplink. The downlink traffic is also reduced. We have $V^{(cache)}_{1,N} < V_{1,N}$, and $V'^{(cache)}_{1,N} < V'_{1,N}$, therefore $C^{tr(cache)}_{1,N} < C^{tr}_{1,N}$.

We further denote $\ddot{v}_i$ and $\ddot{\ddot{v}}_i$ as the amount of requested data that are stored at the level i cache, and the amount of requested data that are served by the cache, respectively. The cost of utilising a cache between i and j is $C^{op(cache)}_i = r_i l_{i,j}\dot{v} + s_i\ddot{v}_i + t'_i\ddot{\ddot{v}}_i$, where r_i represents the cost of processing a request (including cached content searching), s_i is the cost related to the data storage and management, and t'_i is the cost of transmission when the content is provided by the cache, which can be different from $B'_{i,j}$. Including the cost of the caches, we are able to formulate the total cost of data service as $C^{(cache)}_{1,N} = C^{tr(cache)}_{1,N} + \sum_{i=1}^{N-1} C^{op(cache)}_i$. It is observed that in order to achieve a financial benefit for the use of caching, the network should have a reasonably high CHR so that the benefit generated from traffic reduction outweighs the cost overhead of the caches. Cache devices with low data processing and storage costs are always welcomed. In the choice of using the proposed parallel processing strategy, the tradeoff between the reduction of the data access delay and the extra processing cost should be considered. This is because the additional cost $O^{neg}_{i,j}$ is introduced for transmitting and processing the negative request whenever a cache finds the requested file. As a result, C^{op}_i is increased. Besides, since all content requests are still sent to the core network in parallel with the intermediate cache processing, such signalling will increase the uplink traffic.

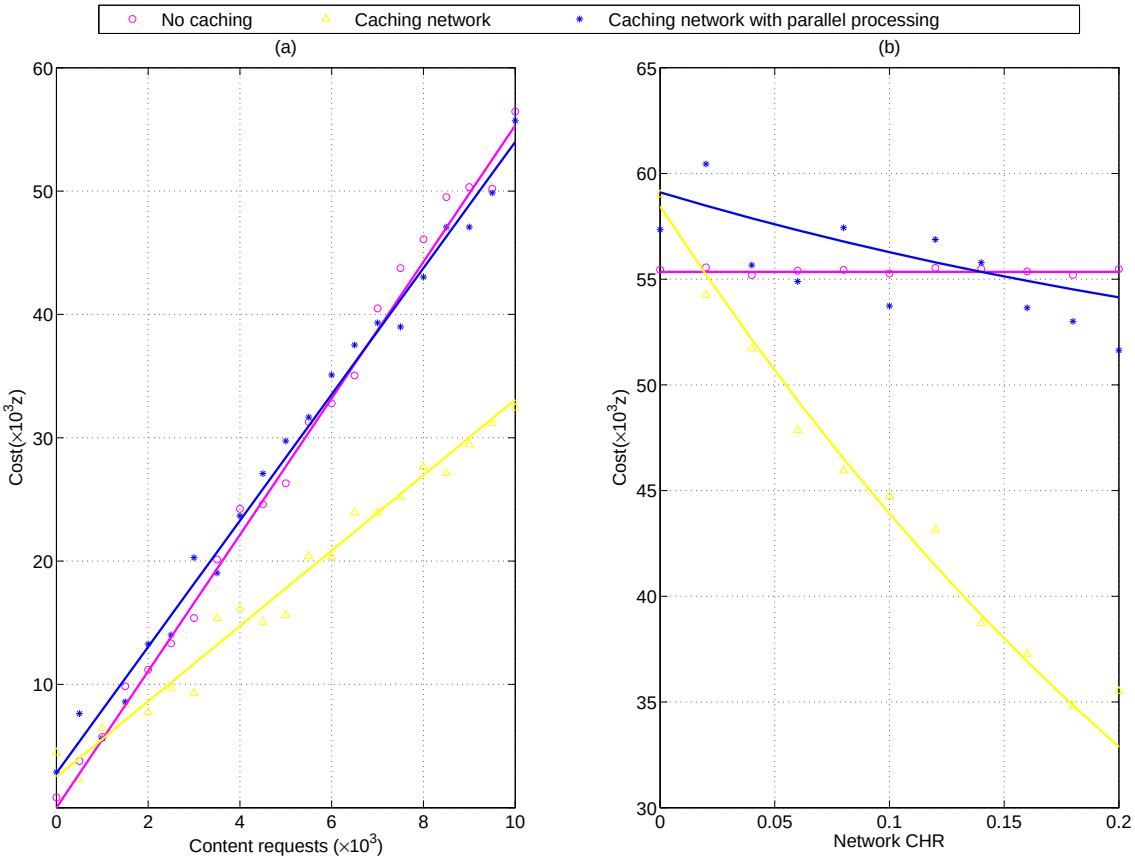

Fig. 4: Cost evaluation of caching networks: (a) Cost as data traffic increases with CHR=0.2; (b) Cost of serving 10^4 data requests with different CHRs.

B. Numerical Evaluation

The costs of processing data requests and data transmission are evaluated using the same simulated networks as before. In addition, we assume that the average size of a content request is 50KB, and the average size of a requested file is 1MB. Define z the unit cost for transmitting one MB data in the downlink from the core network to the end users. In the uplink, the data transmission and processing cost is $20z$ per MB, the cost for the negative request is $10z$. We roughly estimate the cache processing costs as referring to [9] and [14]. The cache hit ratio varies among different layers of the network. We use a small random value to describe the cost variation for transmitting data between different layers considering the different transmission distances and link capacities.

Figure 4(a) shows the cost in different networks as the amount of requests increases. The cost of serving 10^4 requests (around 10GB data transmission) without caching is about $56000z$. In a caching network with a total CHR of 0.2, due to the cost of cache storage, the total cost is higher than that in a non-caching network. As the number of data transmission increases, the benefit appears. The cost reduction reaches approximately a third for serving 10GB data. It is observed that the use of parallel processing results in a cost overhead compared with normal caching network. Such overhead is increasing with the number of requests. Figure 4(b) depicts the tendency of cost for serving 10GB of data with various CHRs. When the CHR is very low, caching networks bring no benefits as they require more data storage and management. As the CHR increases, the ~~cost-benefit~~ in terms of reducing the data transmission is seen. However, the slope of decreasing is relatively small when parallel processing is used. This again reflects the cost for additional processing and signalling at the caches. Only when there are massive content requests, or when the system CHR is reasonably high, parallel processing is more cost-effective than

non-caching network. We conclude that parallel processing will bring extra cost for the caching networks. There is a trade-off between the further reduction of user's content access delay and the additional cost in the choice of using parallel processing. Enabled by more advanced caching managements, we envisage the economic potential of the proposed technique for dense, fast response networks as we are expecting in the foreseeable future.

VI. Conclusions and Future Work

Distributed caching is a promising strategy for wireless cellular networks in order to reduce the content access delay, backhaul data traffic and transmission cost. This article explored the use of caching in the network with a focus on the processing of user requests. We introduced a novel caching strategy using parallel processing and evaluated its performance in terms of reducing the data access delay and redundant traffic in various caching scenarios. We also studied the capital cost for implementing the proposed caching networks. The tradeoff between the performance in terms of delay reduction and the cost overhead needs to be carefully considered in the implementation of the proposed parallel processing. Simulation results have illustrated the effectiveness of the proposed strategy.

We see several opportunities for future research. Firstly, we discussed the benefits of distributed caching assuming that the popular contents have been already stored in the caching devices. In practice, how to dynamically select the caching content out of massive items in the networks is a valuable research topic. Secondly, even if each layer of the network is installed with caches, the issue of how to obtain maximum benefits by using a limited number of caches needs further study. Consider the deployment of cache devices with the installation of new network facilities such as cellular base stations, the optimal selection of locations becomes essential. Cooperative caching techniques have been discussed in [15]. However, huge challenges emerge with the increasingly complicated network topology. For the application on a massive scale, the efficiency of cooperative algorithms is of vital importance. Finally, distributed caching can be regarded as an important component of the emerging fog network technologies that extend the cloud computing paradigm to the edge of the network in order to take full advantages of the more powerful local data processing, transmission, and cooperative resource management in edge devices.

References

[1] Ericsson mobility report *available online: http://www.ericsson.com/res/docs/2015/ericsson-mobility-report-june-2015.pdf*, 2015.

[2] F. Boccardi, R. W. Heath, A. Lozano, T. L. Marzetta, and P. Popovski, "Five disruptive technology directions for 5G," *IEEE Communications Magazine*, vol. 52, pp. 74–80, February 2014.

[3] M. Gregori, J. Gomez-Vilardebo, J. Matamoros, and D. Gunduz, "Wireless content caching for small cell and D2D networks," *IEEE Journal on Selected Areas in Communications*, vol. PP, no. 99, pp. 1–1, 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," *Communications Magazine, IEEE*, vol. 52, pp. 131–139, February 2014.

[5] 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, pp. 82–89, Aug 2014.

[6] M. Ji, G. Caire, and A. F. Molisch, "Fundamental limits of caching in wireless D2D networks," *IEEE Transactions on Information Theory*, vol. 62, pp. 849–869, Feb 2016.

[7] M. Taghizadeh, K. Micinski, S. Biswas, C. Ofria, and E. Torng, "Distributed cooperative caching in social wireless networks," *IEEE Transactions on Mobile Computing*, vol. 12, pp. 1037–1053, June 2013.

[8] N. Abedini and S. Shakkottai, "Content caching and scheduling in wireless networks with elastic and inelastic traffic," *IEEE/ACM Transactions on Networking*, vol. 22, pp. 864–874, June 2014.

[9] J. Erman, A. Gerber, M. Hajiaghayi, D. Pei, S. Sen, and O. Spatscheck, "To cache or not to cache: The 3G case," *IEEE Internet Computing*, vol. 15, pp. 27–34, March 2011.

[10] 3GPP Specifications (Release 10 and beyond), *Available online: http://www.3gpp.org/specifications*.

[11] G. Huston, “Web caching,” *The Internet Protocol Journal*, vol. 2, pp. 1–40, February 1999.

[12] Y. Ma and A. Jamalipour, “A cooperative cache-based content delivery framework for intermittently connected mobile ad hoc networks,” *IEEE Transactions on Wireless Communications*, vol. 9, pp. 366–373, January 2010.

[13] S. Borst, V. Gupta, and A. Walid, “Self-organizing algorithms for cache cooperation in content distribution networks,” *Bell Labs Technical Journal*, vol. 14, pp. 113–125, Fall 2009.

[14] H. Sarkissian, “The business case for caching in 4G LTE networks,” *Available online: http://www.wireless2020.com/docs/LSI_WP_Content_Cach_Cv3.pdf*.

[15] Y. Zhang, H. Qu, and J. Zhao, “A distributed caching based on neighbor cooperation in CCN,” in *Wireless Communications, Networking and Mobile Computing (WiCOM)*, pp. 386–392, Sept 2014.