Northumbria Research Link

www.northumbria.ac.uk/nrl

A Cooperation-Driven ICN-based Caching Scheme for Mobile Content Chunk Delivery at RAN

Xu Zhang* and Yue Cao†
* Department of Computer Science, Xi'an University of Technology, Xi'an, China
Email: zhangxu@xaut.edu.cn
† Department of Computer and Information Sciences, Northumbria University, Newcastle upon Tyne, UK
Email: yue.cao@northumbria.ac.uk

Abstract—In order to resolve the tension between continuously growing mobile users' demands on content access and the scarcity of the bandwidth capacity over backhaul links, we propose in this paper a fully distributed ICN-based caching scheme for content objects in Radio Access Network (RAN) at eNodeBs. Such caching scheme operates in a cooperative way within neighbourhoods, aiming to reduce cache redundancy so as to improve the diversity of content distribution. The caching decision logic at individual eNodeBs allows for adaptive caching, by taking into account dynamic context information, such as content popularity and availability. The efficiency of the proposed distributed caching scheme is evaluated via extensive simulations, which show great performance gains, in terms of a substantial reduction of backhaul content traffic as well as great improvement on the diversity of content distribution, etc.

I. INTRODUCTION

With the evolvement of mobile communication techniques, the adoption of mobile devices becomes widespread. As a result, massive mobile traffic volumes have been generated by users using smart phones and tablets for accessing content objects from the Internet [1], e.g., at Content Delivery Network (CDN) servers or Peer-to-Peer (P2P) source peers. In particular, the significantly increasing demand for content objects have imposed overwhelming stress over the Radio Access Network (RAN) backhaul, which accounts for one of the main bottlenecks in the mobile cellular network [16]. Meanwhile, users are more concerned on the perceived service of experiences, whereas contents are still inefficiently delivered following an host-centric paradigm based on the traditional Internet architecture. Such mismatch between high-quality content demands and unsustainable host-to-host network architecture has made efficient content delivery challenging.

In order to effectively deliver contents and promote feasible distribution of contents in the Internet, Information- or more specifically Content-Centric Networking (ICN/CCN) [2],[3] has recently been introduced to replace the traditional host-to-host communication model. A key feature of ICN/CCN is that content objects can be cached within the network for local access by future interested clients (or the *in-network caching*). With ICN logic, the *naming* of content objects allows for interception of passing-through data items [3], facilitating the content-aware caching at network intermediate nodes. Therefore, coupling ICN with mobile cellular network could be an attractive solution [4][15][27][28], in order to resolve

the tension between the tremendous growth of mobile content traffic and backhaul bottleneck. For instance, it will be of great flexibility to satisfy mobile users' demands locally by caching popular contents at a clustering point, such as at routers or base stations, thereby reducing backhaul traffic and improving overall network capacity.

The potential of utilizing cache space at RAN is expected to satisfy users' demands locally by caching popular contents. Relevant studies exploiting RAN caching at evolved NodeBs (eNBs) have shown great improvement on traffic load reduction [12][22]. And yet, severe duplicate caching over a large number of eNBs could be incurred with existing RAN caching solutions, due to the independency on caching decision making at individual eNBs that intrinsically promotes caching top popular contents with rough similarities. The rational is mainly owing to the fact that only a small portion of very popular contents account for the majority access [5][6]. On the other hand, demand for less popular contents typically accounts for a "long tail" according to measurement studies [6]. Consequently, backhaul transmission could potentially suffer from unexpected content demand increment, incurred by the portion of outbound traffic for retrieving the "long-tail" less popular contents. Therefore, it is imperative to reduce cache redundancy and improve the diversity of content distribution among eNBs. In addition, most previous works exploit caching based on predefined criteria, overlooking the importance of context-awareness, which is closely correlated to traffic demands.

In this paper, we aim to address the aforementioned research challenges by introducing a *fully distributed in-network caching protocol* at eNBs. Specifically, with the objective of assuring the diversity of content distribution, we promote *local coordination* among nearby eNBs to avoid massive duplicate caching across multiple eNBs within a Mobile Network Operator's (MNO) network, through periodical update of caching status with each other. This requires the cooperation among eNBs, which can be realized by eNBs' interconnected high-capacity links, e.g., via X2. The decision-making logic behind such caching operations takes into account important *context information*, such as content popularity, which is measured based on the dynamic changes of user demands. In addition, content availability advertised from original sources is also taken into account in the caching decision-making logic, which

resolves the content unavailable issue especially in a P2P-like system [7][26]. With the cooperative caching protocol, most content demands can be satisfied within a RAN, thereby reducing the expensive backhaul payload. Considering scalability, Bloom filters can be adopted for aggregating information on cached chunks in signalling between coordinated eNBs [23].

II. RELATED WORK

In mobile cellular networks, due to the fast growth of cellular traffic that cannot be coped well with the evolvement of cellular infrastructures, there have been many research works on Web caching at backhaul links [8],[9] or at mobile devices [10],[11]. Caching at RAN have been exploited in [12][22] that mainly focus on independent caching. [12] introduced a caching policy to cache videos at eNBs at the edge of RAN, based on video popularity in a cell. They show that great traffic load can be reduced and users' perceived quality of experiences can be improved at the same time. In [22], a FemtoCaching scheme is proposed and cached contents are predetermined by a centralized base station. Differently, in our work, content caching is adaptive to the dynamic changes of users' demands, and operates in a cooperative way with nearby eNBs, without the help of any centralized controllers. In addition, there are few research works on cooperative eNB caching, and an early work [25] with web caching concerns only exploited cooperative request forwarding in a hierarchical way, the cooperation of which is limited to selected proxy caches at base stations, without considering caching strategies.

As proposed in [4], 5G mobile networks are expected to be equipped with ICN-capable caching schemes at gateways, routers or eNBs, etc., so as to alleviate the aforementioned tension between the mobile data explosion and the backhaul bottleneck. Authors of [14] realized an architecture of adaptive mobile video streaming and sharing in the named data networking (AMVS-NDN), wherein a device-to-device communication mode is triggered if two mobile stations are close enough. A demonstration performed in [15] further proved the effectiveness of introducing ICN into mobile cellular networks, which shows great reduction of content delivery time and traffic load over the mobile backhaul.

Motivated by the above works, we focus on chunk-level distributed in-network caching at eNBs in the RAN for improving both service and operation efficiencies. Specifically, each eNB cooperates with adjacent eNBs for caching decision-making and serving requests not able to be satisfied locally. The proposed cooperative caching protocol is expected to enable efficient content distribution in the cellular network. Details will be provided in the following sections.

III. COOPERATIVE IN-NETWORK CACHING OF CONTENT CHUNKS AT RAN

A. Scheme Overview

Following the ICN paradigm [3], content objects are split into chunks of unified size, and requests are generated at chunk-level. To support the proposed distributed caching protocol, eNBs are required to be equipped with content-aware
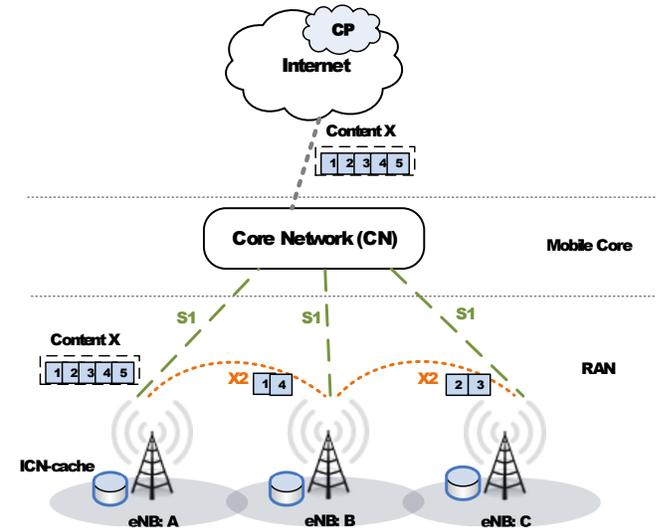

Fig. 1. An example of mobile network architecture with in-network caching capability in the RAN

functionality, so as to be capable of intercepting and caching passing-through content chunks.

Fig. 1 shows a generic architecture of a mobile cellular network, which incorporates cache-enabled eNBs at RAN. Content requests generated from mobile users can be potentially satisfied at local eNBs, instead of traversing through the mobile core network (CN) into the Internet where the specific content provider (CP) is located, thus alleviating the backhaul traffic. Content demands of mobile users can be served at eNBs provided that: 1) the local eNB has cached desired content chunk(s); and 2) the local eNB is aware of any demanded data cached *nearby* at neighboring eNBs. To facilitate the capability of serving requests from nearby eNBs, a cooperative caching scheme needs to be executed at individual eNBs, which can be realized through the X2 interface, given existing works on coordinating interface management via direct X2 links [20]. The purpose of cooperative caching is two-fold: to avoid duplicate chunk caching among nearby eNBs, and also to offer an eNB to explicitly direct its incoming requests to a neighbor without resorting to further remote source CP.

For example in Fig. 1, assume that the content object X is divided into a total number of 5 chunks. Suppose eNB A has cached all chunks of content X. Mobile users subscribing to eNB B and eNB C also request for the same content X, and chunks No. 1, 4 and No. 2, 3 are still missing at cell B and cell C, respectively. Instead of fetching those missing chunks from the original source of CP located in the Internet, the demand could be served locally by nearby eNB A (or eNB B) alternatively, via X2 interfaces.

As presented previously, local context information can be considered for the input of our strategized caching decision-making of mobile content chunks. We further give definitions in the following on the context information and discussion on how such information can be measured and taken into account for making efficient caching decisions.

Let Pop_j denote chunk j's popularity degree, expressed

as $\frac{req_j}{\sum_i req_i}$, given req_j as the number of requests for chunk j received at the eNB. $\sum_i req_i$ is the normalization factor with respect to all request records at the eNB. Let A_j denote chunk j's availability intensity, expressed as $\frac{a_j}{\sum_i a_i}$, given a_j as the number of replicas of chunk j available at source CPs. Thus $(1 - A_j)$ indicates chunk j's unavailability degree. $\sum_i a_i$ is the normalization factor with respect to all availability records at the eNB. Since in the ICN scenario [3], common requests for the same content object can be identified and thus be aggregated at a eNB, the value of req_j can thus be obtained. The value of a_j can be collected at an eNB from content publication originated at source CPs, and any updates of data availabilities at original source CPs are supposed to be intercepted by eNBs during the advertisement originated from CPs [18][21].

It should be noted that the accurate content popularity measurement is challenging due to dynamic changes of users' demands, and thus most current RAN caching with popularity concerns are based on predefined content popularity distribution, which are not adaptive to practice, however. In our work, the popularity is measured based on received requests, collected within a sliding time window, so as to capture the dynamic changes of users' demands. The evaluation of an optimal time window size and sliding time slot is not our focus in this paper, which, however, could be a research point in our future work. A recent research work on content popularity monitoring is proposed in [24], which also shows that accurate content popularity measurement is non-trivial and provided a Bloom filter-based measurement approach, while the false-positive with a Bloom filter is another issue.

With the above context information collected at individual eNBs, an eNBs is configured to be able to cache incoming content chunks in a probabilistic way as the following.

$$Prob_{caching}(j) = Pop_j \cdot (1 - A_j) \qquad (1)$$

Note $Prob_{caching}(j) \equiv 1$ for each in-coming chunk under a universal caching policy that caches *everything everywhere* [3],[13]. With Eq. (1), individual eNBs are able to make caching decision in a strategized way, with higher caching chances on rare content chunks. Note only chunks not cached nearby will be considered for caching locally based on Eq. (1), as will be stated in detail in the next section.

With the top-level overview stated above, the principle of our proposed cooperative in-network caching protocol is as follows, with respect to *cooperative request forwarding* and *cooperative caching*, respectively.

- Each eNB executes a specific *caching algorithm* upon receiving a content chunk, e.g., based on Eq. (1).
- Adjacent eNBs also periodically exchange with each other a list of locally cached items, via a Bloom filter aggregation. And by checking the list, the eNB can simply transmit the received chunk to mobile users without caching it, if the chunk is already cached elsewhere (*cooperative caching*).

- With awareness of the "summary" of already cached chunks nearby, the eNB can forward requests not able to be served locally to nearby eNBs, which could potentially hold the demanded data (*cooperative forwarding*).

B. Cooperative Request Message Forwarding

The *request message* is generated at a mobile user for a specific content chunk demand and forwarded via its attached eNB towards the source. For simplicity, the naming of content objects follow a flat name rule, such as an identifier of the requested content (*content ID*) and the sequence number of the requested chunk (*chunk SN*), which uniquely identify a chunk item and are contained in the request message.

Individual eNBs keep track of content chunk popularity similarly to works like [17][24], based on received requests measured within a sliding time window. Note the time window for popularity collection can be application-dependable. Applications with quite dynamic popularity patterns can be set with a relatively small value on the time window, while how to choose an optimal time window will be left for our future work.

Upon the arrival of a chunk request message, the eNB first updates the chunk popularity information by intercepting the request message, e.g., counting requests during the sliding time window. Then the eNB checks its local cache. In case the requested chunk is cached locally, it can be directly forwarded back to the requester. Otherwise, the eNB then checks whether the requested chunk has been cached at any of its neighbors. This can be enabled by allowing each eNB to locally maintain a "*summary*" of caching status at neighbouring eNBs, which can be achieved by periodically disseminating the local caching status within the neighbourhood, typically through the Bloom filter aggregation. If the desired chunk is found in the summary, an enquiry is made to the neighbor. If a negative ACK (NACK) message is sent back in response to the enquiry (or no relevant information found in the summary), the eNB simply forwards the request to CN towards the original source in the Internet. Otherwise, the request message is forwarded towards the dedicated neighbor. The operation at an eNB when receiving a chunk request message is as shown in Algorithm 1.

Algorithm 1 Cooperative Request Forwarding (req)

1: content ID = get_content_id (req);
2: chunk SN = get_chunk_sn (req);
3: **if** (check_local_cache(content ID & chunk SN) = true) **then**
4:    send_back_chunk_to_requestor(content ID & chunk SN);
5: **else if** (check_neighbor_cache_summary(content ID & chunk SN) = true) **then**
6:    make an enquiry to the nearby eNB for the desired chunk;
7:    **if** (ACK) **then**
8:      send_req_to_neighbor(content ID & chunk SN);

```
 9:      else
10:          forward_req(content ID & chunk SN);
11:      end if
12:   else
13:       forward_req(content ID & chunk SN);
14:   end if
15: end if
```

C. Cooperative Caching of Content Chunks

A *content chunk* to be served to requesting mobile users is generated eigher at a source CP in the Internet, or retrieved from the cache of an eNB, in response to a request message. The content chunk normally contains the content ID and chunk SN, apart from the data itself.

When an eNB receives a content chunk, it first executes specific caching algorithm. With coordinated caching algorithm, the eNB avoids duplicate caching by checking its neighbors' cached items. As discussed previously, a "summary" table maintained locally facilitates such capability, which can be updated when an eNB receives an advertisement message. If none of its neighbors have cached the incoming chunk, the eNB makes caching decision locally, e.g., based on Eq. (1). If the algorithm decides not to cache the chunk, the eNB simply forwards the chunk towards requestors. Otherwise, the eNB inserts the chunk in the cache before forwarding this chunk towards requestors. When adding a chunk to the cache, the eNB needs to execute a cache eviction policy if the cache is fully occupied, such as based on either a Least Recently Used (LRU) or a Least Frequently Used (LFU) replacement policy. The algorithm of cooperative caching at an eNB is given in Algorithm 2.

Algorithm 2 Cooperative Caching among Neighboring eNBs (chunk)

```
 1: content ID = get_content_id (chunk);
 2: chunk SN = get_chunk_sn (chunk);
 3: if (check_neighbor_cache_summary(content ID &
    chunk SN) = true) then
 4:    send_back_chunk_to_requestor(content ID & chunk
       SN);
 5: else
 6:    execute specific caching algorithm;
 7:    if (cache_decision(content ID & chunk SN) = true)
       then
 8:       cache the chunk;
 9:    else
10:       send_back_chunk_to_requestor(content  ID  &
          chunk SN);
11:    end if
12: end if
```

IV. SIMULATION RESULTS

In this section, we use extensive simulations to evaluate the efficiency of the proposed caching policy. For comparison purpose, the non-caching scenario as well as the native independent-universal caching policy is considered [3],[13].

A. Simulation Setup

We conduct the simulation in NS-3. A simple RAN topology consisting of 7 eNBs is considered. eNBs are randomly interconnected with each other and are all connected to one CN node. There are 10 content objects considered in the experiment, and each has a unique content identifier, the content ID. Each content is further divided into 100 chunks of unified size, which are distinguished (within the content) by a unique sequence number, SN. A total number of 10'000 requests are generated at chunk-level, uniformly distributed at eNBs. The popularity distribution of content objects follows the Mandelbrot-Zipf distribution [19] with parameter $\alpha = 0.7$ and $q = 50$. Regarding the chunk-level popularity, a uniform random distribution is considered within each content scale. eNBs are cache-enabled with cache size defined in terms of the number of chunks. And the LRU replacement policy is adopted under each caching algorithm.

B. Performance Metrics

Simulation results are collected based on the following metrics in order to evaluate the caching performance.

1) *Cache Hit Ratio*: The proportion of chunk requests served at eNBs.

2) *Cache Replacement Ratio*: At a single eNB, the cache replacement ratio is calculated as the proportion of incoming chunks that can trigger cache evictions. The average value among all eNBs is taken.

3) *Cache Redundancy Ratio*: The intensity of a duplicate caching among neighbouring eNBs in RAN, which can be calculated as the proportion of the overall cache capacities in RAN that hold duplicate data item.

4) *Backhaul Traffic Intensity*: The proportion of chunk requests that have to traverse through CN towards original CPs.

5) *Source Access Delay Improvement*: Compared to non-caching scenario, the improvement of access delay to demanded data.

C. Results

We compare in Fig. 2 the cache hit ratio under different caching policies with the increment of cache size. As observed, overall the performance is increasing with the cache space's increment under all caching polices. Specifically, the proposed caching policy outperforms greatly the native universal caching policy. With cooperative forwarding & caching capabilities, the proposed caching scheme achieves the best performance, while the native independent-universal caching policy performs the worst due to its content-blind caching. Note that the proposed policy with independent caching still achieves better cache hit ratio than the universal policy, which is equipped with capabilities of cooperative forwarding & caching. The rational regarding the desirable performance of the proposed caching scheme is due to its intelligently selective caching decision-making logic at eNBs, rather than a blindly aggressive caching. In order to further evaluate the efficiency of the proposed caching policy, we compare the
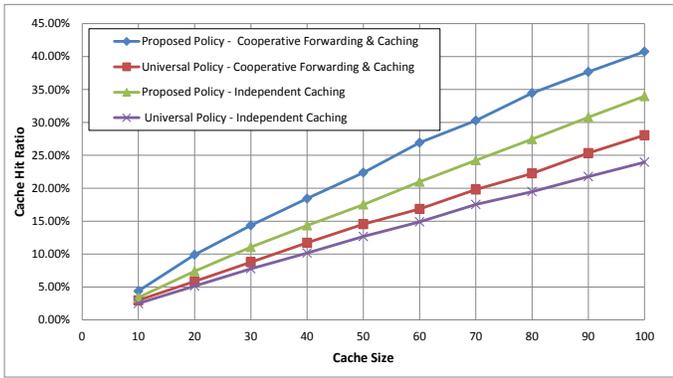
Fig. 2.   A comparison of cache hit ratio under different caching policies
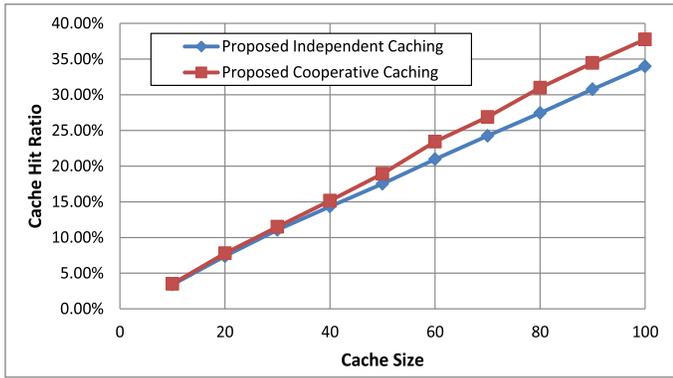

Fig. 4.   A comparison of cache replacement ratio under different caching policies

Fig. 3.   A comparison of cache hit ratio under different caching policies

Fig. 5.   A comparison of cache redundancy ratio under different caching policies

Fig. 6.   RAN backhaul traffic intensity under different caching policies

cache hit ratio in Fig. 3 under the proposed caching policy, with two different operational scales: the independent caching and the cooperative caching (without cooperative request forwarding), respectively. The purpose of this comparison is to show whether a solely cooperative caching without request redirection is still able to achieve desirable caching performance, which is especially important in the scenario when the communication overhead is a big concern. As shown in the figure, even without cooperative request forwarding scheme, the cooperative caching under our policy is able to achieve desirable cache hit ratio.

Fig. 4 shows the cache replacement ratio under different caching policies. Not surprisingly, the proposed caching policy outperforms the universal caching policy significantly. In particular, the proposed cooperative forwarding & caching policy reduces cache replacements by more than 60% at cache size 100, compared to the massive replacements under the native-independent universal caching policy. Additionally, the proposed independent policy still outperforms the cooperative-universal policy greatly. The reduced cache replacements can alleviate the operational cost at individual eNBs, saving electricity power consumption and thus achieving a certain energy-efficiency. Furthermore, low number of cache replacements implies an increased overall network capacity, since more content demands can be served locally at eNBs.
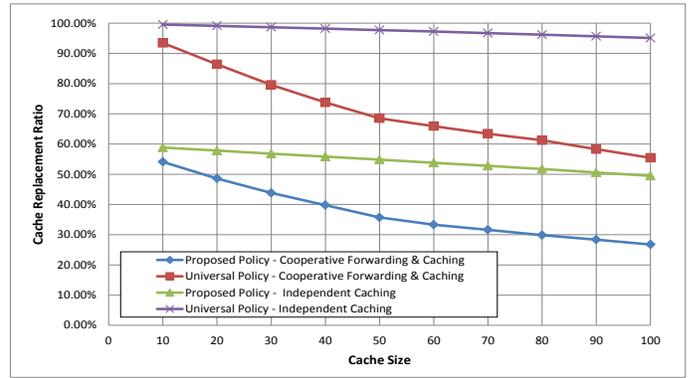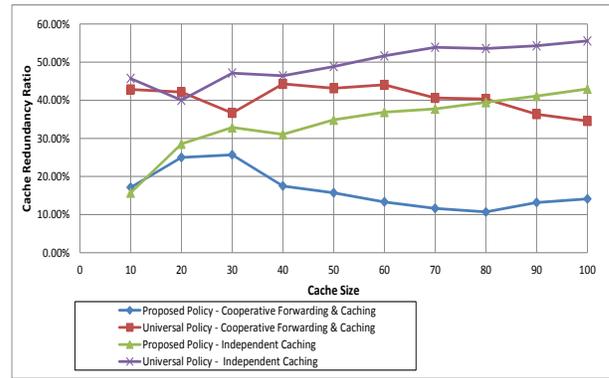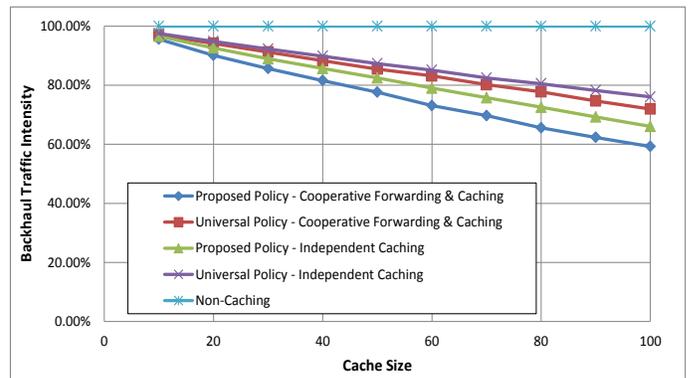
The cache redundancy ratio is evaluated in Fig. 5. As can be seen, the proposed cooperative forwarding & caching policy exhibits the best performance, while the independent-universal caching policy performs the worst, due to its aggressive caching nature. In particular, the cache redundancy can be reduced by more than 35% under the proposed caching policy, as compared to the independent universal caching. The results explain the importance of promoting cooperative caching among neighboring eNBs, so as to achieve diverse distribution of content chunks in the network.
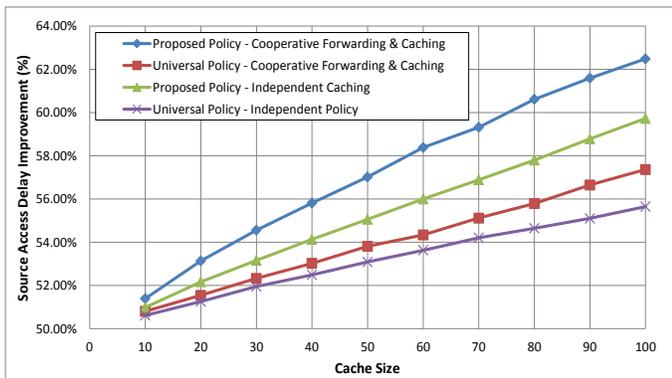
Fig. 7. Source access delay improvement (%) under different caching policies

With in-network caching deployed at RAN, great backhaul traffic can be reduced significantly as shown in Fig. 6. In particular, more than 20% of backhaul traffic load can be reduced under a native independent-universal caching policy with cache size bigger than 70. And even more backhaul traffic can be alleviated under the proposed independent caching policy. Further with cooperation promoted among nearby eNBs, up to 40% of backhaul traffic load can be reduced at cache size 100.

From mobile users' performance concern on the access delay to fetch the demanded content objects, we evaluate the source access delay improvement in Fig. 7. As observed, the access delay can be reduced by at least 50% under a native in-network caching policy, in comparison to the non-caching scenario. Further reduction of more than 60% can be achieved with cache size at 100, under the proposed cooperative caching policy.

As observed from the above results, with ICN-capable in-network caching at eNBs in RAN, significant mobile backhaul traffic can be reduced. In particular, under an intelligently selective caching policy, the performance from both network's and users' sides can be further improved significantly.

V. Conclusion

We focus on the issue of explosive backhaul traffic in the mobile cellular network. In order to reduce the risk of bottleneck over RAN backhaul, we proposed a fully distributed in-network caching protocol executed at individual eNBs in RAN. In order to improve diverse content distribution to serve more demands from mobile users, the proposed caching scheme supports a cooperative caching logic. With extensive simulation performed, the results show that the proposed caching policy is able to reduce significant backhaul traffic volumes and at the same time, mobile users' access delay can be reduced considerably.

References

[1] X. Wang et al., *A Survey of Green Mobile Networks: Opportunities and Challenges*, ACM/Springer Mobile Networks and Applications (MONET), vol. 17, no. 1, 2012, pp. 4-20.
[2] T. Koponen et al., *A Data-Oriented (and beyond) Network Architecture*, SIGCOMM, vol. 37, no. 4, pp. 181-192, 2007.
[3] Van Jacobson et al., *Networking Named Content*, Proc. of the 5th international conference on Emerging networking experiments and technologies, ACM CoNEXT'09, 2009.
[4] Xiaofei Wang et al., *Cache in the Air: Exploiting Content Caching and Delivery Techniques for 5G Systems*, IEEE Communication Magazine, Feb., 2014.
[5] S. Woo et al., *Comparison of Caching Strategies in Modern Cellular Backhaul Networks*, ACM MobiSys, June, 2013.
[6] B. A. Ramanan et al., *Cacheability Analysis of HTTP Traffic in an Operational LTE Network*, Wireless Telecommun. Sysmp., Apr., 2013.
[7] Daniel S. et al., *Content Availability and Bundling in Swarming Systems*, IEEE/ACM Tran. on Networking, Vol. 21, No. 2, 2013.
[8] J. Erman et al., *To cache or not to cache: The 3G case*, IEEE Internet Computing, Mar., 2011.
[9] J. Erman et al., *Over the Top Video: The Gorrilar in Cellular Networks*, in Proc. of the ACM SIGCOMM, 2011.
[10] F. Qian et al., *Web Caching on Smartphones: Ideal vs. Reality*, In Proc. of the ACM MobiSys, 2012.
[11] H. Chen et al., *Cache Access and Replacement for Future Wireless Internet*, IEEE Communication Magazine, vol. 44, no. 5, 2006.
[12] Hasti Ahlehagh et al., *Video Caching in Radio Access Network: Impact on Delay and Capacity*, IEEE Wireless Communications and Networking Conference (WCNC), 2012.
[13] E. J. Rosensweig and J. Kurose, *Breadcrumbs: efficient, best-effort content location in cache networks*, in INFOCOM, 2009.
[14] B. Han et al., *AMVS-NDN: Adaptive Mobile Video Streaming with OFFloading and Sharing in Wireless Named Data Networking*, IEEE INFOCOM, NOMEN Worshops, 2013.
[15] Diego Perino et al., *A High Speed Information-Centric Network in a Mobile Backhaul Setting*, 1st ACM Conference on Information-Centric Networking (ICN), 2014.
[16] *Mobile operators risk backhaul gap in LTE networks*, http://www.telecomstechnews.com/news/2013/feb/13/mobile-operators-risk-backhaul-gap-in-lte-networks/, 2013.
[17] Sem Borst et al., *Distributed Caching Algorithms for Content Distribution Networks*, in Proc. of IEEE INFOCOM, 2010.
[18] Cohen, B., *Incentives build robustness in BitTorrent*, in Proceedings of P2P Economics Workshop, Berkeley, CA, USA, 2003.
[19] Z. Silagadze, *Citations and the Zipf-Mandelbrot's law*, Complex Systems, vol. 11, no. 487-499, 1997.
[20] S. Rangan and R. Madan, *Belief propagation methods for intercell interface coordination in femtocell networks*, IEEE J. Sel. Areas Commun., vol. 30, no. 3, pp. 631-640, Mar., 2012.
[21] Erik Nygren et al., *The Akamai Network: A Platform for High-Performance Internet Applications*, ACM SIGOPS Operating Systems Review, vol. 44, no. 3, July 2010.
[22] N. Golrezaei et al., *FemtoCaching: Wireless Video Content Delivery through Distributed Caching Helpers*, IEEE INFOCOM, Mar. 2012.
[23] A. Broder et al., *Network Applications of Bloom Filters: A survey*, Internet Mathematics, pp. 636-646, 2002.
[24] Huichen Dai et al., *Towards Line-Speed and Accurate On-line Popualrity Monitoring in NDN Routers*, in Proc. IEEE International Symposium of Quality of Service (IWQoS), 2014.
[25] Z. Zhang et al., *Cooperativve Proxy Caching for Wireless Base Stations*, in Proc. Mob. Inf. Syst., vol. 3, no. 1, IOS, 2007.
[26] X. Zhang et al., *A Distributed In-Network Caching Scheme for P2P-like Content Chunk Delivery*, in Proc. Computer Networks, Elsevier, Vol. 91, No. 14, 2015.
[27] Cao, Yue, et al., *An electric vehicle charging management scheme based on publish/subscribe communication framework*, IEEE Systems Journal, 2015.
[28] Cao, Yue, et al., *Routing in delay/disruption tolerant networks: A taxonomy, survey and challenges*, IEEE Communications surveys & tutorials vol. 15, no. 2, 2013.