Caching-as-a-Service: Virtual Caching Framework in the Cloud-based Mobile Networks

Xiuhua Li, Xiaofei Wang, Chunsheng Zhu, Wei Cai, and Victor C. M. Leung
Dept. Electrical and Computer Engineering, The University of British Columbia, Vancouver, Canada
Email: {lixiuhua, xfwang, cszhu, weicai, vleung}@ece.ubc.ca

***Abstract*—Over recent years, the demand for rich multimedia services over mobile networks has been soaring at a tremendous pace. However, it is envisioned that traditional dedicated networking equipment in mobile network operators (MNOs) cannot support the phenomenal growth of the traffic load and user demand dynamics, but consume unnecessary energy resource inefficiently. The emerging techniques for mobile content caching and delivery become more and more attractive, by which popular content can be cached inside mobile front-haul and back-haul networks, so that demands to the same content from users in proximity can be easily accommodated without redundant transmissions from the remote resource, thereby eliminating duplicated traffic significantly. While the incorporation between advanced cloud computing technologies and network function virtualization (NFV) techniques has become an essential issue in the evolution process of mobile systems, in this article, we propose the concept of "Caching-as-a-Service" (CaaS), a caching virtualization framework along with the development of Cloud-based Radio Access Networks (C-RAN), and the virtualization of Evolved Packet Core (EPC). Then we study the potential techniques related to the cache virtualization, and discuss technical details of caching virtualization and system optimization for CaaS. We carry out numerical evaluation on proposed framework and show significant improvement on the performance of reducing inter-MNO traffic load and intra-MNO traffic load.**

***Index Terms*—Virtual Caching, Content Delivery Network, Virtualzation, Future Mobile Networks.**

I. Introduction

Along with recent advances in mobile communication technologies, an ever-growing amount of mobile users are continuously attracted to enjoy a wide plethora of multimedia services using smart phones and tablets [1]. But the capacity of the wireless link, the radio access networks (front-haul), and the core networks (back-haul) cannot practically cope with the explosively growing mobile traffic due to the centralized nature of mobile network architectures. Therefore, revolutionary approaches involving new mobile network architectures and advanced data transmission technologies are anticipated, towards the next generation cloud-based mobile networks. A potential key penetrating point is to cache popular contents at mobile network edges to reduce the traffic due to redundant downloads [2], [3].

With the content caching inside mobile front-haul and back-haul networks due to the "Power Law" effect of the content popularity [2], service demands of mobile users in proximity for the same contents, can be easily accommodated without redundant transmissions from remote resource outside the mobile operator networks, so that duplicated traffic load can be eliminated significantly. However, effective in-network caching in legacy MNO architecture is not practically realizable, due to the dedicated signal processing hardware at RAN and the sophisticated controlling and processing units at EPC.

Recently, the new trend of virtualizing mobile network functions into software-based cloud servers has been envisioned, in order to optimize the resource utilization and thus to reduce expenditures of MNOs. The Software-Defined Networking (SDN) [4] and Network Function Virtualization (NFV) [5] represent a first concrete step towards this direction, catalyzing the idea of decoupling software defined control plane from the underlying hardware-driven data plane, where the hardware is emulated on general purpose servers; the mobile network functions run over the emulated hardware as if it is running on its own bare-metal resources. By virtualization, MNOs can dynamically adjust their functions within virtual machines (VMs) in an online manner easily, while providing quick and elastic services to mobile users, and 3rd-party service providers (SPs) and content providers (CPs) in an on-demand manner. This hence illustrates the new concepts of the "Radio-Access-as-a-Service (RAaaS)", which runs all signal processing units in VMs back in the data center but connects multi-RAT (Radio Access Techniques) antennas by high-capable fiber for offering high flexibility for radio access functions [6], and the "EPC-as-a-Service (EPCaaS)", which also virtualizes the EPC functions into VMs in the cloud in order to be managed elastically [5].

Therefore, the virtualization of mobile networks motivates us a new potential research direction, ***the virtualized caching inside MNOs' cloud center***, which in this article, for the first time to the best of our knowledge, is proposed as "Caching-as-a-Service (CaaS)". Rather than running traditional Content Delivery Network (CDN) services virtually, which are still statical storage of files and management units in VMs of the cloud, e.g., MetaCDN [7] and ActiveCDN [8], CaaS instances in the mobile clouds can be adaptively created, immigrated, scaled (up or down), shared and released depending on the user demands and requirements from 3rd-party SPs; caching in CaaS is more flexible. By CaaS, MNOs can provide highly flexible and programmable virtual caching capability to 3rd-party SPs, in order to serve mobile users with highly qualified and customized services. MNOs also need to ensure that the mobile network equipments are used more productively with necessary traffic optimization, task scheduling and load balancing techniques inside its cloud centers, transparent to

978-1-4673-7131-5/15/$31.00 ©2015 IEEE

mobile users and 3rd-party SPs.

The remainder of this paper is organized as follows. After studying the related work in Sec. II, we discuss the infrastructure modeling of virtual caching in Sec. III. Then we discuss two optimization frameworks for the virtual caching resource in Sec. IV and Sec. V, respectively. The performance evaluation is shown in Sec. VI, and the conclusion along with discussion on potential research issues is in Sec. VII.

II. Related Work

Recent research emphasis turns to virtualize the traditional radio access processing functions into the cloud, so called "C-RAN" technique [6], regarding the concept of "Radio-Access-as-a-Service", with remote antennas, say Remote Radio Units (RRUs), connected with the servers running the virtualized Baseband Processing Units (BBUs) in the MNO's data center by high-speed front-haul fiber networks. Moreover, MNOs are also designing the decomposition of EPC into virtual environments of clouds, with the full feature compatibility of native EPC [5] with regards to the concept of "EPC-as-a-Service". So the trend of cloud-based mobile network virtualization unveils a new paradigm of cloud-based cooperative cell caching. Different from theoretical studies for caching videos in base stations e.g., FemtoCache in [13] and video caching in [12], which lack practical implementation consideration, and those for simply running CDN services in the cloud, e.g., MetaCDN [7] and ActiveCDN [8], which need to be elaborated more for mobile networks, "Caching-as-a-Service" has more advantages on flexibility and elasticity, which will be discussed in following sections.

III. Infrastructure Modeling

A. Cache Virtualization and "CaaS"

We first illustrate the CaaS framework with three conceptual layers along with the RAaaS and EPCaaS in the emerging mobile networks as shown in Fig. 1: ***Layer 1:*** - The physical hardware servers in an MNO's data center are running the VMs of RAN and EPC as well as caching. ***Layer 2:*** - The caching VMs (cVMs) coexist with those of RAN and EPC, where since everything is virtualized, the caching can be considered as the universal caching discussed in [3]. For example, if we cache a content at an eNB, we will update the path (routing table) for the file from the cVMs to the RAN VMs, or even we can directly immigrate a whole cache VM into the server in a better position with less network congestions. In this layer, contents can be chunked, replicated, distributed, bundled, and redirected freely among cVMs, depending on the realistic traffic dynamics, content popularity, and the diversity of user demands. ***Layer 3:*** Each 3rd-party SPs can programme with the virtual caching by CaaS's application programming interfaces (APIs). Caching policies and strategies can be static or dynamic, online or offline. 3rd-party SPs can optimize their cVMs with regard to virtual network topology and transmission paths. Also MNO can dynamically charge for the resource utilization of 3rd-party SPs.

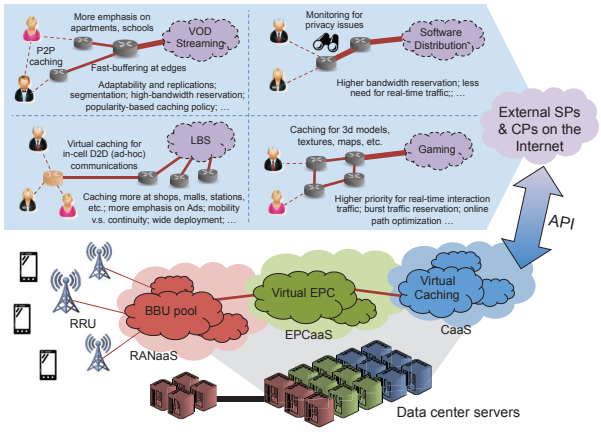

Fig. 1. Illustration of "Caching-as-a-Service"

The virtual caching based on mobile cloud computing can realize the cVMs anywhere inside MNO's system with properly allocated positions in an appropriate distance to the mobile users at the edge, and to the involved RAN and EPC functions, according to the requirements from mobile users, to the preserved Quality of Service (QoS) guarantees for 3rd-party SPs, and to the constraints by the physical facilities. As illustrated in Fig. 2, cVMs can be attached and immigrated from any servers to any other ones in order to achieve global optimal scheduling. The caching here is quite universal and can take place anywhere due to the mixture of cVMs with RAN VM and EPC VMs, since the BBU pools, EPC core networks, and caching servers are within the same data center infrastructure, or in individual data centers but connected by high-capable inter-connections. Furthermore, any content in cVMs can be chunked (divided) into more pieces, or will be prepared in packet formats, so that fast content delivery may happen directly via servers running RAN VMs to users without flowing back to EPC VMs. A centralized caching controller (possibly based on SDN) plays a more important role in this scenario, but due to limited space, we will further elaborate the system design in our future study.

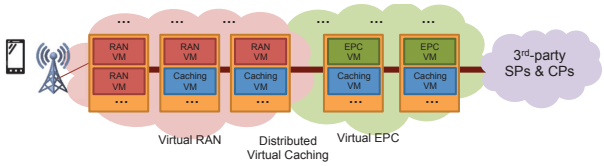

Fig. 2. An illustration of the deployment of caching virtual machines

B. System Modeling

A structural system model of VM caching is shown in Fig. 3 [9]. Outside the MNO, some 3rd-party SPs provide the contents over the Internet, while inside the MNO, there are massive cells covering the whole service area. Besides, each cell is associated to a specific cVM and each cVM has certain caching functionality and storage. All the cVMs effectively share their caching information to achieve efficient cooperation among cVMs inside the mobile network. Each cVM is able

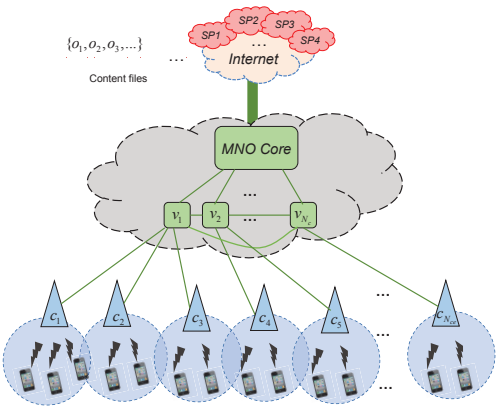

Fig. 3. System structural modeling

to decide how to effectively select popular contents to cache based on the content popularity in the cVM's connected cells. Moreover, the cache files need to be updated in time according to the changes of demands from mobile users, in order to improve the last-mile service quality since the files are mostly near mobile users.

TABLE I
MODELING PARAMETERS AND NOTATIONS

Symbol	Definition
N_c	Total number of cVMs in the network
N_{ce}	Total number of cells in the network
N_o	Total number of contents in the network
$\mathcal{C}_i$	The set of cells connected to cVM i
v_i	cVM i
c_n	Cell n
o_j	Content j
S_i	Size of cache storage of cVM i
s_j	Size of content j
x_{ij}	A decision 0-1 variable for cVM i to cache content j
p_{lj}	Popularity at cell n for content j
r_{ij}	Popularity at cVM i for content j

In the mobile network, there are totally N_c cVMs with finite cache sizes and N_o contents with various popularity in different N_{ce} cells. Besides, cVM i ($i \in \{1, 2, \ldots, N_v\}$) is connected with a set of cells, denoted by $\mathcal{C}_i \subseteq \{1, 2, \ldots, N_{ce}\}$. The caching strategy in the mobile network is denoted by a binary matrix $\mathbf{X} := (x_{ij})_{N_c \times N_o} \in \{0,1\}^{N_c \times N_o}$, where $x_{ij} = 1$ means that content j is cached in the cVM i while $x_{ij} = 0$ means no caching. The more detailed modeling parameters and notations are given in Table I. Besides, we have $r_{ij} = \sum_{n \in \mathcal{C}_i} p_{nj}$. Moreover, following [10], we assume that the overall popularity of the content j, denoted as $P_j = \sum_{i=1}^{N_c} r_{ij}$ for $\forall j$, satisfies the Zipf distribution. Specifically, we have $P_j = \frac{rank_j^{-\beta}}{\sum_{j=1}^{N_o} rank_j^{-\beta}}$, $\forall j$, where $rank_j$ is the rank of the content j in the descending order of content popularity, and the exponent $\beta \in (0, 1]$ is a constant. Thus, we have $\sum_{i=1}^{N_c} \sum_{j=1}^{N_o} r_{ij} = \sum_{j=1}^{N_o} P_j = 1$.

Moreover, to carry out effective caching, all the optimization objectives, i.e., reducing the expected sum inter-MNO traffic load (external) or reducing the expected sum intra-MNO traffic load (internal traffic) in the schemes, will be focusing on finding the optimal binary matrix $\mathbf{X}$ with various constraints and considerations. Besides, for the strategy of no caching, the MNO network's expected sum traffic load can be calculated as $TL_0 = N_o N_c \sum_{i=1}^{N_c} \sum_{j=1}^{N_o} r_{ij} \cdot s_j$.

IV. REDUCING INTER-MNO (EXTERNAL) TRAFFIC LOAD

Considering the inter-domain routing and transmissions between Internet 3rd-party SPs, one 3rd-party SP needs to always pay for the exchanged traffic to other 3rd-party SPs. So MNO should carefully design the caching policy to shrink the inter-MNO bandwidth payment, which instead indicates that the MNO needs to optimally cache contents with a high diversity so that any content demand from the users can be satisfied within network formed by cVMs and cells in the MNO. That is, we need to increase the cache hit ratio within the MNO. Thus, the 1st optimization objective is to minimize the expected sum traffic load that goes out of the MNO, which is instead to maximize the satisfaction of content demands from all users within the MNO, either locally by the cVMs which the users' attached cells is connected to, or by other cVMs. Based on the storage limitation of cVMs, the optimization problem about reducing expected sum inter-MNO traffic load can be formulated as

$$\max N_o N_c \left\{ \sum_{i=1}^{N_c} \sum_{j=1}^{N_o} x_{ij} \cdot r_{ij} \cdot s_j + \sum_{i=1}^{N_c} \sum_{j=1}^{N_o} \left[\left(\bigcup_{k=1}^{N_c} x_{kj} - x_{ij} \right) \cdot r_{ij} \cdot s_j \right] \right\} \tag{1a}$$

$$s.t. \quad \sum_{j=1}^{N_o} x_{ij} \cdot s_j \leq S_i, \quad \forall i, \tag{1b}$$

$$\bigcup_{k=1}^{N_c} x_{kj} = x_{1j} \oplus x_{2j} \oplus \ldots \oplus x_{N_c j}, \quad \forall j, \tag{1c}$$

$$x_{ij} \in \{0, 1\}, \quad \forall i, \forall j. \tag{1d}$$

Moreover, the first part of the optimization objective in (1a) is for the content deliveries locally satisfied by the cVMs which the users' attached cells are connected to and its second part is for the content deliveries satisfied by remote cVMs because no local cVM may have the required contents. It is obvious that in order to increase the diversity of the cached contents, one content file can be only cache with only one copy within the MNO, and hence the caching storage can be effectively utilized to be able to store more contents. So we have an additional constraint, that is to cache one copy, or not to cache, $\sum_{i=1}^{N_c} x_{ij} \leq 1, \forall j$. Furthermore, the objective in (1a) becomes $N_o N_c \sum_{i=1}^{N_c} \sum_{j=1}^{N_o} \left(\bigcup_{k=1}^{N_c} x_{kj} \right) \cdot r_{ij} \cdot s_j =$

$\sum_{i=1}^{N_c}\sum_{j=1}^{N_o} x_{ij}\cdot\left(N_oN_c\sum_{k=1}^{N_c} r_{kj}\cdot s_j\right) = \sum_{i=1}^{N_c}\sum_{j=1}^{N_o} x_{ij}\cdot R_j$, where $R_j = N_oN_c\sum_{k=1}^{N_c} r_{kj}\cdot s_j, \forall j$. Thus, the problem in (1) can be rewritten as

$$\max\ Z_1 = \sum_{i=1}^{N_c}\sum_{j=1}^{N_o} x_{ij}\cdot R_j \tag{2a}$$

$$s.t.\ \sum_{j=1}^{N_o} x_{ij}\cdot s_j \le S_i,\quad \forall i, \tag{2b}$$

$$\sum_{i=1}^{N_c} x_{ij} \le 1,\quad \forall j, \tag{2c}$$

$$x_{ij}\in\{0,1\},\quad \forall i,\forall j, \tag{2d}$$

which is a regular 0-1 multi-knapsack problem (MKP) [11]. Besides, for this caching strategy, the MNO network's expected sum traffic load can be calculated as $TL_1 = TL_0 - Z_1$, and the cache hit ratio by using this strategy can be calculated as $h_1 = \frac{1}{N_o}\sum_{i=1}^{N_c}\sum_{j=1}^{N_o} x_{ij}$.

A polynomial-time approximate algorithm [11] is used to achieve the caching scheme for reducing inter-MNO traffic load, which works as follow. The contents and cVMs are sorted such that $\frac{R_1}{s_1}\ge\frac{R_2}{s_2}\ge\ldots\ge\frac{R_{N_o}}{s_{N_o}}$ and $S_1\le S_2\le\ldots\le S_{N_c}$.

Based on the above sorting for contents and cVMs, an initial feasible solution is determined by applying the greedy algorithm to the first cVM, then to the second one by using the only remaining items, and so on. This is obtained by calling N_c times the following procedure shown in Algorithm 1, giving the the size $\overline{S}_i = S_i$ of the current cVM and the current solution, of value z, stored, for $\forall j$, in

$$y_j = \begin{cases} 0, & \text{if content } j \text{ is currently unassigned,} \\ \text{index of the cVM it is assigned to,} & \text{otherwise.} \end{cases} \tag{3}$$

Obviously, for $\forall j$, if $y_j \ne 0$, then $x_{y_jj} = 1$; otherwise, $x_{y_jj} = 0$. Then the polynomial-time approximate algorithm's procedure is shown in Algorithm 2, which takes $O(N_oN_c)$ time in total.

Algorithm 1 Greedy Algorithm.

Input: N_o, (R_j), (s_j), z, (y_j), i, $\overline{S}_i$.
Output: z, (y_j), $\overline{S}_i$.
for $j = 1$ to N_o **do**
  if $y_j = 0$ and $s_j \le \overline{S}_i$ **then**
    $y_j := i$, $\overline{S}_i := \overline{S}_i - s_j$, $z := z + R_j$.
  end if
end for

V. Reducing Intra-MNO (Internal) Traffic Load

In order to maintain the network devices for supporting the MNO's traffic load, it is also important to reduce the intra-MNO transmissions in order to lower the cost due to deployment and maintenance of cables and transport devices in the MNO. So the 2nd optimization objective is to minimize the expected sum traffic load within the MNO's network, which

Algorithm 2 Polynomial-time Approximate Algorithm.

Input: N_o, N_c, (R_j), (s_j), (S_i).
Output: Z_1, (y_j).
—Procedure 1. [Initial Solution]
$z := 0$, $y_j := 0$ for $\forall j$, $\overline{S}_i := S_i$ for $\forall i$.
for $i = 1$ to N_c **do**
  Call Greedy Algorithm in Algorithm 1.
end for
—Procedure 2. [Rearrangement]
$z := 0$, $\overline{S}_i := S_i$ for $\forall i$, $i := 1$.
for $j = N_o$ to 1 **do**
  if $y_j > 0$ **then**
    Let l be the first index in $\{i,\ldots,N_c\}\cup\{1,\ldots,i-1\}$ such that $s_j \le \overline{S}_l$.
    if no such l **then**
      $y_j := 0$.
    else
      $y_j := l$, $\overline{S}_l := \overline{S}_l - s_j$, $z := z + R_j$.
      if $l < N_c$ **then**
        $i := l + 1$.
      else
        $i := 1$.
      end if
    end if
  end if
end for
for $i = 1$ to N_c **do**
  Call Greedy Algorithm in Algorithm 1.
end for
—Procedure 3. [First Improvement]
for $j = 1$ to N_o **do**
  if $y_j > 0$ **then**
    for $k = j + 1$ to N_o **do**
      if $0 < y_j \ne y_j$ **then**
        $h := \arg\max\{s_j, s_k\}$.
        $l := \arg\min\{s_j, s_k\}$.
        $d := s_h - s_l$.
        if $d \le \overline{S}_{y_l}$ and $\overline{S}_{y_h} + d \ge \min\{s_u : y_u = 0\}$ **then**
          $t := \arg\max\{R_u : y_u = 0 \text{ and } s_u \le \overline{S}_{y_h} + d\}$.
          $\overline{S}_{y_h} := \overline{S}_{y_h} + d - s_t$, $\overline{S}_{y_l} := \overline{S}_{y_l} - d$, $y_t := y_h$, $y_h := y_l$, $y_l := y_t$, $z := z + R_t$.
        end if
      end if
    end for
  end if
end for
—Procedure 4. [Second Improvement]
for $j = N_o$ to 1 **do**
  if $y_j > 0$ **then**
    $\overline{S} := \overline{S}_{y_j} + s_j$, $\mathcal{Y} := \emptyset$.
  end if
  for $k = 1$ to N_o **do**
    if $y_k = 0$ and $s_k \le \overline{S}$ **then**
      $\mathcal{Y} := \mathcal{Y}\cup\{k\}$, $\overline{S} := \overline{S} - s_k$.
    end if
  end for
  if $\sum_{k\in\mathcal{Y}} R_k > R_j$ **then**
    $y_k := y_j$ for $\forall k \in \mathcal{Y}$, $\overline{S}_{y_j} := \overline{S}$, $y_j := 0$, $z := z + \sum_{k\in\mathcal{Y}} R_k - R_j$.
  end if
end for
$Z_1 \leftarrow z$.
—Procedure 5. [Recovery]
Sorting (y_j) and recover its elements' indexes according to the original orders.

is to maximize the user demands that can be satisfied locally by the cVMs which the users' attached cells is connected to. The corresponding problem can be formulated as

$$\max\ Z_2 = N_o N_c \sum_{i=1}^{N_c}\sum_{j=1}^{N_o} x_{ij}\cdot r_{ij}\cdot s_j \tag{4a}$$

$$s.t.\ \sum_{j=1}^{N_o} x_{ij}\cdot s_j \le S_i,\quad \forall i, \tag{4b}$$

$$x_{ij}\in\{0,1\},\quad \forall i,\forall j, \tag{4c}$$

which is a generalized assignment problem (GAP) [11]. In this problem, to satisfy user demands locally as much as possible, one content can be cached in multiple cVMs, i.e., $\sum_{i=1}^{N_c} x_{ij}$ is not limited. Besides, for this caching strategy, the MNO network's expected sum traffic load can be calculated as $TL_2 = TL_0 - Z_2$, and the cache hit ratio by using this strategy can be calculated as $h_2 = \frac{1}{N_o}\sum_{j=1}^{N_o}\bigcup_{i=1}^{N_c} x_{ij}$.

This problem can be split into N_c independent regular 0-1 single knapsack subproblems, the i-th subproblem of which can be expressed as

$$\max\ z_i = N_o \sum_{j=1}^{N_o} x_{ij}\cdot r_{ij}\cdot s_j \tag{5a}$$

$$s.t.\ \sum_{j=1}^{N_o} x_{ij}\cdot s_j \le S_i, \tag{5b}$$

$$x_{ij}\in\{0,1\},\quad \forall j, \tag{5c}$$

Clearly, we have $Z_2 = \sum_{i=1}^{N_c} z_i$. Greedy algorithm is also used for solving the subproblems in parallel and its procedure is shown in Algorithm 3 [11] based on the sorting as $r_{i1}\ge r_{i2}\ge\ldots\ge r_{iN_o}, \forall i$. Solving each subproblem takes $O(N_o)$ time, so the total time of solving this problem is $O(N_c N_o)$.

Algorithm 3 Greedy Algorithm for Solving Subproblems.

Input: N_o, i, $(r_{ij})_{j=1}^{N_o}$, (s_j), S_i.
Output: z_i, $(x_{ij})_{j=1}^{N_o}$.
$\overline{S} := S_i$, $z_i := 0$, $j^* := 1$.
for $j = 1$ to N_o **do**
  if $s_j > \overline{S}$ **then**
    $x_{ij} := 0$.
  else
    $x_{ij} := 1$, $\overline{S} := \overline{S} - s_j$, $z_i := z_i + r_{ij} * s_j$.
  end if
  if $r_{ij} * s_j > r_{ij^*} * s_{j^*}$ **then**
    $j^* := j$.
  end if
end for
if $r_{ij^*} * s_{j^*} > z_i$ **then**
  $z_i := r_{ij^*} * s_{j^*}$, $x_{ij} := 0$ for $\forall j$, $x_{ij^*} := 1$.
end if
$z_i \leftarrow N_o N_c \cdot z_i$.
Sorting (x_{ij}) and recover its elements' indexes according to the original orders.

VI. Evaluation Results

In this section, we mainly evaluate the network expected sum traffic load and cache hit ratio performance of the strategies of caching contents in cVMs and no caching. In our simulation, the cache sizes of the cVMs are equal to a constant $S^{\max}$, that is, $S_i \equiv S^{\max}$ for $\forall i$. The overall popularity of each content (P_j) follows Zipf distribution and the popularity of contents (r_{ij}) is random in $(0,1]$. Note that for a given P_j $(\forall j)$, the random values r_{ij} $(\forall i,\forall j)$ should satisfy $\sum_{i=1}^{N_c} r_{ij} = P_j$ by simply setting $r_{ij} \leftarrow \frac{r_{ij}}{\sum_{i=1}^{N_c} r_{ij}}\cdot P_j$ [14]. Besides, the contents' sizes are random in $[0.001, 1]$ Gbits. To simplify the description, we denote the strategies of optimization for inter-MNO traffic load and intra-MNO traffic load as Scheme 1 and Scheme 2, respectively.

Fig. 4, Fig. 5 and Fig. 6 compare the expected sum traffic load performance of the non-caching strategy, Scheme 1 and Scheme 2, as well as compare the cache hit ratio performance of Scheme 1 and Scheme 2 based on different considerations.

From Fig. 4(a), for $N_c = 10$ cVMs with the cache size $S^{\max} = 50$ Gbits, all the expected sum traffic load of the considered caching/non-caching strategies goes up linearly with the increase of the contents' number. Specifically, the proposed Scheme 1 and Scheme 2 can effectively reduce the expected sum traffic load, and Scheme 1 greatly outperforms Schemes 2 since users' content demands can be satisfied by remote MNO's cVMs while the demands can not in Scheme 2. Besides, as the exponent β increase, the traffic load achieved by Scheme 1 and Scheme 2 decreases, which agrees with the Zipf distribution's properties. From Fig. 4(b), the cache hit ration with the two schemes decreases with the increase of the contents' number because of the limitation of cVMs' storage, and Scheme 1 can get higher cache hit ratio than Scheme 2 since Scheme 1 allows to cache at most one copy for each content within the MNO while multiple copies can be cached in Scheme 2. Besides, the value of β has little effect on the cache hit ratio.

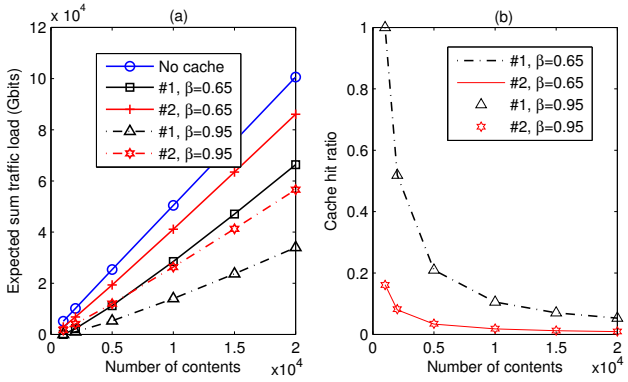

Fig. 4. Traffic load and cache hit ratio versus number of contents in $N_c = 10$ cVMs with $S^{\max} = 50$ Gbits.

From Fig. 5(a), for fixed $N_o = 5,000$ contents with the cache size $S^{\max} = 50$ Gbits, as the cVMs' number increases, the expected sum traffic load of the non-caching strategy and Scheme 2 increases linearly, while the traffic load in Scheme 1

gradually goes to zero since the increase of total cache storage of cVMS can cache more contents and even all the considered contents. As well, the proposed Scheme 1 and Scheme 2 can effectively reduce the expected sum traffic load, and Scheme 1 greatly outperforms Schemes 2. Besides, as the exponent β increases, the traffic load achieved by Scheme 1 and Scheme 2 decreases. From Fig. 5(b), Scheme 1 can even cache all the contents when the number of cVMs is large enough and greatly outperforms Scheme 2 in terms of cache hit ratio. Besides, the value of β also has little effect on the cache hit ratio.

From Fig. 6(a), for fixed $N_c = 50$ cVMs and $N_o = 5,000$ contents, the expected sum traffic load of the non-caching strategy is a constant. As the total cache size (percentage to total content size) increases, the expected sum traffic load of Scheme 1 and Scheme 2 reduces since more contents are cached in cVMs. Also, the proposed Scheme 1 and Scheme 2 can effectively decrease the expected sum traffic load, and Scheme 1 has much better performance than Scheme 2. Besides, the traffic load in Scheme 1 and Scheme 2 decreases as the exponent β increases. From Fig. 6(b), with the increase of the total cache size, the cache hit ratio in Scheme 1 significantly increases while that in Scheme 2 gradually increases, which also shows that Scheme 1 outperforms Scheme 2. Also, the value of β has little effect on the cache hit ratio.

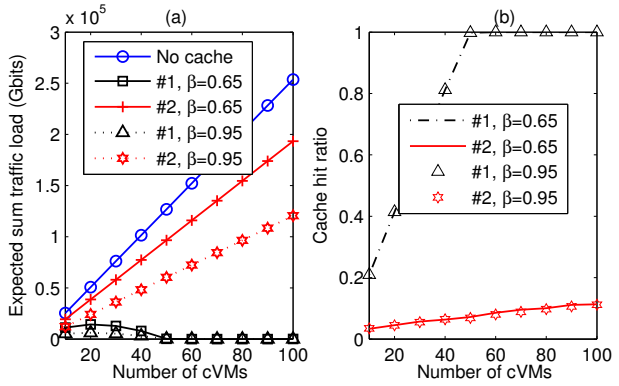

Fig. 5. Traffic load and cache hit ratio versus number of cVMs with fixed $N_o = 5,000$ contents, where $S^{\max} = 50$ Gbits.

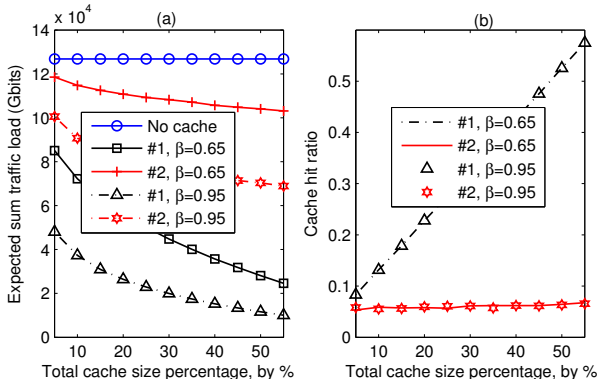

Fig. 6. Traffic load and cache hit ratio versus total cache size (percentage to total content size) in fixed $N_c = 50$ cVMs with fixed $N_o = 5,000$ contents.

VII. Conclusion and Discussion

In this paper, we have proposed CaaS, the virtual caching framework for cloud-based mobile networks, aiming to minimize the inter-MNO and intra-MNO traffic load. We have provided heuristic algorithms to solve the optimization problems from engineering perspective. And numerical evaluation results have shown that CaaS can greatly decrease the network traffic. Focusing on the design complexity for forwarding local traffic and enabling local caching at small cells in real networks [9], more studies on integrating content caching with resource virtualization of radio access networks and EPC systems for next generation mobile systems will be carried out. From the system design perspective, it is also solicited to implement high scalable and programmable CaaS APIs, while testing them practically, in terms of reliability, scalability and complexity. Besides, the demands for more sophisticated mobile broadband services will drive the development of mobile networks with new potential issues: 1) prefetching and caching for mobile video streaming services at the very edge servers and cVMs; 2) scalable cVM allocation and immigration for software distribution services; 3) mobility-based virtual caching with content persistency for location-based services, even with support of device-to-device communications.

Acknowledgment

This work is support in part by a Four Year Doctoral Fellowship of China Scholarship Council, and by the Canadian Natural Sciences and Engineering Research Council through grants RGPIN-2014-06119 and RGPAS-462031-2014.

References

[1] CISCO, "Cisco Visual Networking Index : Global Mobile Data Traffic Forecast Update , 2013 – 2018," Technical Report, 2014.
[2] S. Woo, et al., "Comparison of Caching Strategies in Modern Cellular Backhaul Networks," in *ACM MobiSys*, pp.319–332, June 2013.
[3] X. Wang, M. Chen, T. Taleb, A. Ksentini, 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.
[4] X. Jin, et al., "SoftCell: Scalable and Flexible Cellular Core Network Architecture," in *ACM CoNEXT*, December, 2013.
[5] H. Hawilo, A. Shami, M. Mirahmadi, and R. Asal, "NFV: State of the Art, Challenges and Implementation in Next Generation Mobile Networks (vEPC)," *IEEE Network Magazine*, November, 2014.
[6] China Mobile Research Institute, "C-RAN: The Road Towards Green RAN," *White Paper*, Version 2.5, Oct. 2011.
[7] J. Broberg, "MetaCDN: Cloud Computing Meets Content Delivery," *Journal of Network and Computer Applications*, vol.32, 2009.
[8] S. Srinivasan, "ActiveCDN: Cloud Computing Meets Content Delivery Networks," *Columbia University*, Tech. Rep., 2012.
[9] X. Wang, et al., "A Framework of Cooperative Cell Caching for the Future Mobile Networks," *in HICSS*, Jan. 2015.
[10] M. Cha, H. Kwak, P. Rodriguez, Y. Y. Ahn, and S. Moon, "I Tube, You Tube, Everybody Tubes: Analyzing the World's Largest User Generated Content Video System," *In Usenix/ACM SIGCOMM IMC*, Oct. 2007.
[11] S. Martello, and P. Toth. Knapsack Problems: Algorithms and Computer Implementations. John & Sons, Inc. New York, USA, 1990.
[12] H. AhleHagh and S. Dey, "Video caching in Radio Access Network: Impact on delay and capacity," in *WCNC*, pp.2276–2281, April 2012.
[13] N. Golrezaei, K. Shanmugam, A.G. Dimakis, A.F. Molisch and G. Caire, "FemtoCaching: Wireless Video Content Delivery through Distributed Caching Helpers," in *INFOCOM*, pp.1107–1115, March 2012.
[14] X. Li, X. Wang, S. Xiao, and V. C.M. Leung, "Delay performance analysis of cooperative cell caching in future mobile networks", in *ICC 2015*, to appear.