Content Distribution for 5G Systems Based on Distributed Cloud Service Network Architecture

Lirong Jiang, Gang Feng and Shuang Qin
National Key Laboratory of Communications, University of Electronic Science and Technology of China
Chengdu, 611731, China
[e-mail: lrjiang@std.uestc.edu.cn]
*Corresponding author: Lirong Jiang

Received February 4, 2015; revised May 14, 2015; revised July 12, 2015; accepted August 27, 2015; published November 30, 2015

Abstract

Future mobile communications face enormous challenges as traditional voice services are replaced with increasing mobile multimedia and data services. To address the vast data traffic volume and the requirement of user Quality of Experience (QoE) in the next generation mobile networks, it is imperative to develop efficient content distribution technique, aiming at significantly reducing redundant data transmissions and improving content delivery performance. On the other hand, in recent years cloud computing as a promising new content-centric paradigm is exploited to fulfil the multimedia requirements by provisioning data and computing resources on demand. In this paper, we propose a cooperative caching framework which implements State based Content Distribution (SCD) algorithm for future mobile networks. In our proposed framework, cloud service providers deploy a plurality of cloudlets in the network forming a Distributed Cloud Service Network (DCSN), and pre-allocate content services in local cloudlets to avoid redundant content transmissions. We use content popularity and content state which is determined by content requests, editorial updates and new arrivals to formulate a content distribution optimization model. Data contents are deployed in local cloudlets according to the optimal solution to achieve the lowest average content delivery latency. We use simulation experiments to validate the effectiveness of our proposed framework. Numerical results show that the proposed framework can significantly improve content cache hit rate, reduce content delivery latency and outbound traffic volume in comparison with known existing caching strategies.

Keywords: Caching, cooperation, content distribution, cloudlet, mobile network

This work has been supported by National High-tech R&D Program of China (No.2014AA01A703), National Science and Technology Fund (No. 61401076), and the Huawei Technologies Co.Ltd HIRP Funding (No.20140108).

http://dx.doi.org/10.3837/tiis.2015.11.001 **ISSN : 1976-7277**

1. Introduction

With the explosive growth of mobile Internet, as well as the extensive deployment of efficient fast-rate 4G/LTE, the demand for high-speed data applications, such as mobile multimedia services, has been extensively emerging up in recent years. We can expect that the further growth of smart mobile devices and mobile data traffic will have enormous impact on the next generation mobile communication system (5G). According to the survey of Cisco [1], mobile data traffic has been increasing at a high speed over the few years, while the total volume of mobile data traffic will rise 13-fold in 2017, compared with that of 2012. The habits of mobile users are mainly focused on multimedia service, such as online video, online music, and P2P streaming sharing. The increasing demand for mobile multimedia services brings a big challenge to future mobile communication system [2]. Thus it is imperative to develop efficient mechanisms for supporting mobile multimedia and data services.

In mobile cellular networks, duplicate downloads of a few popular multimedia contents with large sizes (*e.g.* popular music files, picture files *etc.*) have been observed through experimental studies [3][4]. The redundant download traffic occupies an important portion of mobile multimedia traffic. Therefore, researchers have been investigating effective techniques to reduce the duplicate content transmissions by adopting intelligent caching strategies in cellular mobile networks [3-6]. **Fig. 1** shows a general mobile cellular network architecture equipped with caching capability, where core network and ratio access network (RAN) are the two tiers envisioned for deploying caches due to the all-IP nature of current cellular networks [7]. In this architecture, distributed service data storage at caching enabled core network [3][4] or access network [5][6] can effectively reduce the outbound (remote) traffic volume and response latency to fetch a content file. Specifically, caching in 3G mobile networks [3] and caching in 4G/LTE [8] networks have both been proven to significantly improve content delivery efficiency and performance. Furthermore, it is apparent that efficient caching strategy would enhance the energy efficiency of mobile networks, contributing to the evolution of green 5G networks effectively.

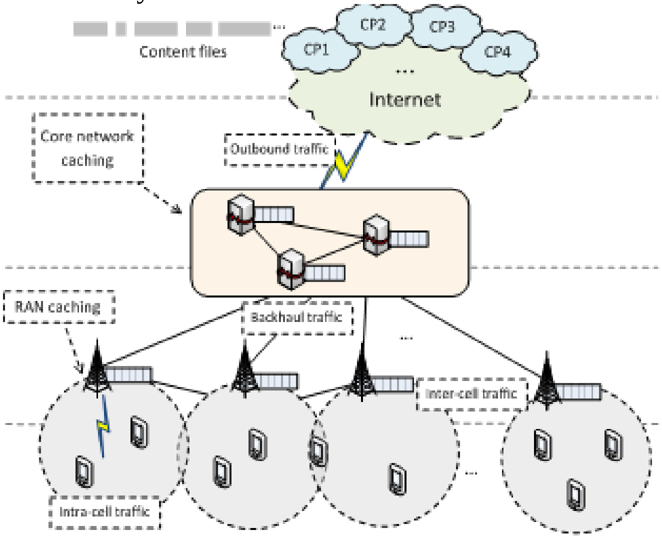


Fig. 1. General mobile network incorporating caching

Using content distribution network (CDN) to distribute data content in local caches close to mobile users allows a user to gain access to the nearest content requested, so as to reduce duplicate data transmissions and content delivery latency. In [27], the authors have investigated the capability of cache enabled content distribution in wireless ad-hoc networks. The concept of resource management for in-network caching environments is introduced and cache management for information-centric networks is studied which tries to reduce caching redundancy and make more efficient use of available cache resources in [28]. The authors in [29] have proposed a comprehensive overview on the area of energy-aware common content distribution over wireless networks with mobile-to-mobile cooperation. In [30], researchers investigate the algorithms and address the stability problem for content distribution over multiple multicast trees. Furthermore, The effectiveness of content distribution can be improved by introducing multi-level (hierarchical) content caching [7][9]. However, we cannot simply apply traditional CDN-based content distribution technique to mobile networks. It is difficult to meet dynamic needs of the users in mobile networks, as legacy CDN based content distribution mechanism is generally designed for traditional wired communication network architecture. In mobile networks, the resources (storage, bandwidth, computing capacity, *e.g.*) and the position of the deployed servers are constrained. More importantly, the hit rate of cached contents could be rather low in mobile networks due to the content dynamics, user mobility and limited number of users in a cell. On the other hand, the scale of content provided by Content Providers (CPs) is growing rapidly and it is thus impossible to cache all contents, although storage cost is becoming much cheaper. Therefore, it is of vital importance to develop efficient caching strategies to enjoy the benefits of local content caching.

On the other hand, in recent years the rapid rise of cloud computing services and the wide deployment of cloud facilities such as Data Centers (DCs) provide new ways to support mobile multimedia and data services. In a recent study of cloud-assisted service computing for future fifth generation (5G) mobile ad-hoc networks in [26], the authors have proposed a mechanism to solve excessive energy consumption in cloud data centers without re-searching and re-link routing for the lost cloud data server by updating link loss information and sending the location of queried data kept in the content map. In [41], a hierarchical cloud computing architecture is proposed to enhance performance by adding a mobile dynamic cloud formed by powerful mobile devices to a traditional general static cloud. Cloud computing has valuable features, such as high scalability, flexibility and on-demand supply, which can well address the needs of content distribution. To facilitate better multimedia and data distribution services by introducing cloud computing, Distributed Cloud Service Network (DCSN) architecture is proposed to provide low-delay and high-quality local cloud services for mobile users [10][11]. DCSN is composed of multiple cloudlets which are widely distributed in mobile users' local areas, and are equipped with strong computing resource and storage space. The resources of cloudlets can be used by nearby mobile devices via one-hop to access [10]. The original studies of applying cloudlets to wireless networks are focused on traditional wireless LAN [10][11][12]. Later, researchers further combine cloudlets with traditional Base Stations (BSs) to provide local cloud services for users in mobile cellular networks [2], where the proposed cloud architecture consists of distributed multimedia DCs, and integrated cloudlet and base stations. The inherent layered and distributed feature of DCSN makes it natural to implement caching to bring about lower content delivery latency and improved user experience.

In the area of 5G, a new network architecture based on Network Function Virtualization (NFV) and Software-Defined Networking (SDN) becomes the prevailing view worldwide [33]. The 5G network will be built upon SDN, NFV and cloud computing technologies, leading to more flexible, intelligent, efficient, and open network system. The 5G network architecture consists of three clouds: access cloud, control cloud, and forwarding cloud. The access cloud supports multiple radio access technologies, integrates the centralized and distributed architectures, and adapts to all sorts of backhaul links, in order to realize more flexible deployment and more efficient radio resource management. The control and data forwarding functions of 5G network will be substantially decoupled, turning into the centralized and unified control cloud and the flexible and efficient forwarding cloud [34][35]. The proposed DCSN architecture can be deemed as an abstract network model of 5G mobile networks with features of cloudification and vitalization. We exploit the inherent layered and distributed feature of DCSN and propose a cooperative caching strategy, taking the advantage of distributed cloud with strong computing resource and storage space to address the increasing demand for mobile multimedia and data services in emerging 5G systems.

Traditional ratio access network architecture faces various challenges in the 4G era and beyond. C-RAN, which is proposed by the China Mobile, is a new type of RAN architecture to help operators address the challenges [39]. C-RAN is responsible for the mobile users' access to networks, and is composed of high-density Remote Radio Heads (RRHs) and centralized Baseband Units pools (BBUs). It is not only applicable to existing mobile networks but also an essential element for future 5G systems [40]. C-RAN is supposed to be able to accommodate and facilitate several 5G technologies such as Large Scale Antenna Systems (LSAS), full duplex, ultra-dense networks, etc., mainly thanks to its inherent centralization natures well as the edibility and scalability of a cloud-based implementation [41].At the access level, C-RAN is responsible for providing efficient large-capacity access function. Meanwhile, at the service level, DCSN is proposed to provide local cloud-based multimedia services to mobile users. To accommodate the massive number of mobile devices and improve the utilization of limited resources, the next generation mobile communication system (5G) should employ a new architecture by introducing C-RAN and DCSN to provide better content delivery services for mobile users. By introducing C-RAN instead of traditional Base Station, and integrating C-RAN with DCSN together, DCSN cooperates with C-RAN to improve the resource utilization efficiency of both wireless resource of C-RAN and caching resources of DCSN, as well as reduce the content access delay to ensure user QoE.

Thus far, although an amount of existing studies are focused on the design and implementation of content distribution mechanisms in mobile cellular networks, most of them are based on heuristic algorithms without analytical modelling and optimal strategy design. Caching strategy, deciding what to cache and how to manage caches, are crucial for overall content distribution performance. There have been a number of caching strategies proposed, such as least recently used (LRU) [13] and least frequently used (LFU) [14], for Internet Web caching. In [15], the authors combine LRU and LFU together and propose the segmented LRU strategy. Randomized replacement (RR) [16] is a cache replacement strategy, which tries to reduce the complexity of the replacement process by using randomized decisions to find an object for replacement. However, these existing works consider only single cache case. Obviously, for the case with multiple caches and they may perform in a cooperation way, existing caching strategies/algorithms cannot be applicable. Intuitively, the cooperation among multiple distributed caches may provide improvement for content distribution. However, it is much more challenging to design cooperative caching strategy to appropriately improve cache performance [7].

In this paper, we propose a cooperative caching framework which implements State based Content Distribution (SCD) algorithm based on DCSN architecture, where several distributed cloudlets with local caches cooperate and share contents with each other. We use content popularity and content state which is determined by content requests, editorial updates and new arrivals to formulate a content distribution optimization model. Data contents are deployed in local cloudlets according to the optimal solution to achieve the minimal average content delivery latency. We conduct simulation experiments to validate the effectiveness of our proposed SCD algorithm, and numerical results indicate that the proposed solution can effectively increase cache hit rate, reduce outbound traffic and content delivery latency compared with existing caching strategies.

The paper is structured as follows. In Section 2 we propose a cooperative caching architecture based on DCSN architecture and develop the framework. Then we describe the content state transition which is used for content distribution modelling in details in Section 3. In Section 4 we derive the SCD model and present the optimal content distribution algorithm. In Section 5 we evaluate the performance of our proposed algorithm, in terms of cache hit rate, content delivery latency and data traffic volume in comparison with LRU, LFU and RR and finally conclude this paper in Section 6.

2. Cooperative Caching Architecture and Framework

Our proposed Hierarchical Cloud Service Network model captures the main attributes of cloudification and virtualization. It is mainly composed of three parts: Core Cloud Server Networks, Distributed Cloudlets Service Networks, and Cloud Radio Access Networks. Core Cloud Server Networks distribute the content (*e.g.*, multimedia) serving ability to Cloudlets which are widely deployed at the edge of networks. Once these local Cloudlets have cached content contents and deployed virtual machines (VMs) in advance, they can provide high-quality and low-delay local services for mobile users. Moreover, we consider to deploy Cloud Radio Access Networks (Cloud-RAN) into this Hierarchical Cloud Service Networks. Cloud-RAN is composed of high-density Remote Radio Heads (RRHs) and centralized Baseband Units pools (BBUs), putting all the signal processing units and wireless resources into BBUs pools by employing virtualization technology, and scheduling these resources based on the load status by employing real-time cloud-computing centralized scheduling methods [36][37]. The cloudlet should allocate appropriate processing, caching space and VM to cache content data for users from remote DCs if local Cloudlet has not pre-cached these contents. However, if Cloudlets have pre-cached these services, users can enjoy the low-delay and high-quality local multimedia services.

In this section, we first describe the cooperative caching architecture for mobile cellular network based on Distributed Cloudlets Service Networks (DCSN), and then present the framework of our proposed State based Content Distribution (SCD) algorithm.

2.1 Cooperative Caching Architecture

In recent years, cloud computing service has been introduced into mobile communication networks as a promising means for providing improved mobile multimedia service. In [10], the authors propose the Distributed Cloudlets Service Networks (DCSN) architecture, which is composed of multiple widely-distributed local cloudlets, to provide low-delay and high-quality local cloud service. The authors of [2] introduce DCSN into the mobile cellular networks, where cloudlets could be deployed with traditional Base Stations (BSs).

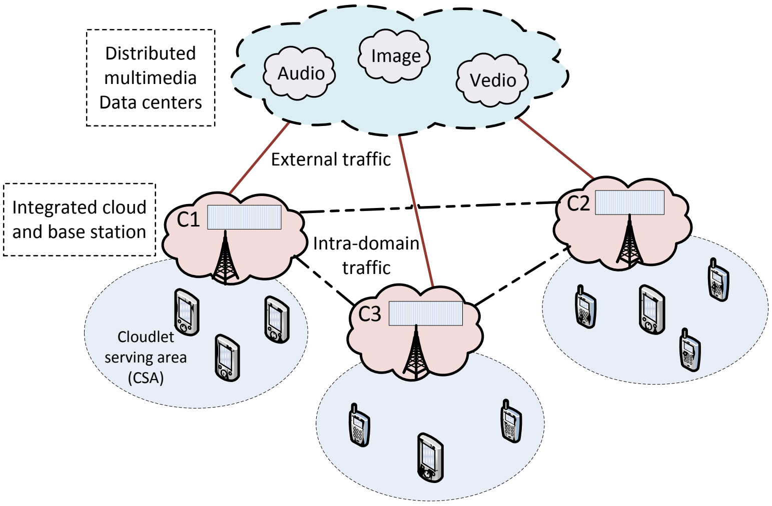

Distributed multimedia Data centers
Audio
Image
Vedio
External traffic
Integrated cloud and base station
C1
C2
C3
Intra-domain traffic
Cloudlet serving area (CSA)

Fig. 2. A cooperative caching domain based on DCSN architecture

Based on DCSN, we propose a caching architecture, as shown in **Fig. 2**, where several local caches of distributed cloudlets may collaborate with each other, to support mobile content service in future mobile networks. The cooperative cloudlets in a region, which could be co-located with base stations, form a cooperative caching domain. One cloudlet is in charge of providing multimedia services for a medium-sized area, called Cloudlet Serving Area (CSA), as shown in **Fig. 2**. The essential idea is that a cloudlet will handle local content requests, collect information about editorial updates and new arrivals in DCs, and decide what and how to cache. Cloudlets in a cooperative caching domain work in a cooperative way to realize content sharing. For example, a content item miss in C1 in **Fig. 2** may be recovered from C2 or C3. DCs may proactively deploy content in local cloudlets. The transfers between DCs and cloudlets are performed typically during off peak hours, using gentle protocols, *e.g.*, one that yields more congestion than TCP and/or using low priority channels, *e.g.*, worse than best effort in DiffServ [17].

The cloudlets are supposed to be deployed and co-located with base stations in mobile cellular networks. To work in a cooperative way, these cloudlets can exchange information (*i.e.* periodically exchange their information about cached content to each other, or share content by fetching data from other cloudlets) by using the communications between the traditional base stations. Indeed, in current 4G/LTE networks, we can use the X2 interface which provides the data transfer function between NodeBs to realize the communications [25]. The X2 interface transfers PDU in user plane using GTP-U and provides reliable signal transmission in control plane using SCTP based on IP protocol. 3GPP has specified X2 interface standard including the frame structure and interface, but leave the implementation of communications to individual equipment vendors [31]. In practice, X2 communication is usually supported by optical communications [32]. Hence it is reasonable to assume that communication capability between BSs in next generation (5G) mobile is supported.

In the proposed caching architecture, we combine content caching and distribution with DCSN, in which the cloud computing technique provides more sufficient resource to support mobile content service for future 5G mobile networks. We exploit the distributed and layered architecture of DCSN to implement content caching and distribution. The cloudlets cooperate with each other, providing better cache performance. To the best of our knowledge, there is no research work to address the optimal caching strategy by using analytical modeling for mobile cloud service networks, aiming at 5G architecture. Our work will benefit the design of content distribution mechanisms and contribute to the evolution of future 5G mobile networks.

2.2 Popularity

The difference of request rate for different content items could be great in mobile communication networks. Due to the different favour of data content for mobile users in different areas, we can use relative popularity to characterize how often a specific content item is requested. The more times a content item requested by mobile users within a unit time, the greater request rate and popularity. A common model characterizing content popularity in this context is the Zipf model or variants thereof [18][19][20]. Let there be Ω content items in the system, and we sort the content items according to the decreasing order of their popularity in a certain Cloudlet Serving Area (CSA). For the rth (r=1,2,…, Ω) most popular content item, the relationship between request rate $p(r)$ and r is as follows: $p(r)$ is proportional to r^{-1} in the pure Zipf model [18], to $r^{-\beta}$ (where β is a strictly positive integer) in the modified Zipf model [19] and to $(r+k)^{-\beta}$ (where k is a strictly positive integer) in the Zipf-Mandelbrot model [20].

We consider a cellular network with N cloudlets, denoted by $\mathbf{C} = \{C_1, C_2, \dots, C_N\}$, which are deployed in a cooperative caching domain. DCs provide Ω content items, namely $\mathbf{O} = \{O_1, O_2, \dots, O_\Omega\}$, which are subject to U editorial updates per day and A new arrivals per day on average. A cloudlet is assumed to receive R requests per day on average. Let λ_n^i denote the popularity of O_i in C_n's CSA. Sorting $\mathbf{O} = \{O_1, O_2, \dots, O_\Omega\}$ according to the decreasing order of λ_n^i in C_n yields $\mathbf{M_n} = \{M_{1,n}, M_{2,n}, \dots, M_{\Omega,n}\}$. Then an one-to-one mapping $\mathbf{O} \overset{f}{\leftrightarrow} \mathbf{M_n}$ is formed in C_n:

$$f\colon \; O_i \leftrightarrow M_{r,n} \qquad (i, r = 1,2, \dots, \Omega),$$

which means that O_i is the rth popular content in C_n's CSA. We will use the mapping later in the design of content distribution strategy. Obviously the mapping relationship could be different in each CSA.

2.3 State Based Content Distribution Framework

Based on our proposed cooperative caching architecture, we develop a State based Content Distribution (SCD) framework, as illustrated in **Fig. 3**. Our design objective is to minimize average total content delivery latency for all users in a cooperative caching domain by deriving an optimal content distribution. Then content items can be pre-allocated to cloudlets according to the optimal distribution solution. When a user requests a specific content item, the local cloudlet will send it to the requesting user if the requested content item is found. Otherwise, the local cloudlet fetches the requested content from either other cloudlets or DCs. Besides, CPs may update the content or insert new content items in DCs. Data content will be periodically (*e.g.* by day or hours) re-distributed based on content state, which is determined by content requests, editorial updates and new arrivals.

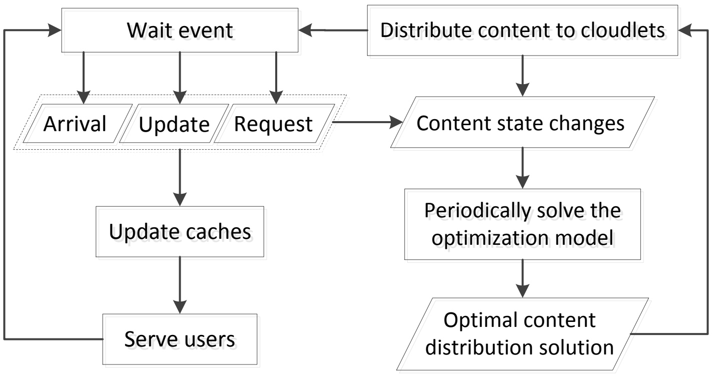

Wait event
Distribute content to cloudlets
Arrival
Update
Request
Content state changes
Update caches
Periodically solve the optimization model
Serve users
Optimal content distribution solution

Fig. 3. The framework of SCD

In the proposed framework, the data content items in DCs are distributed to the cloudlets in a cooperative caching domain according to an optimal solution when the caches are initialized. Corresponding action would be taken to adjust the local caches if any of the events, namely new content arrivals, content updates and user requests, occurs during a specified period, to provide appropriate content services to requesting users. The state of a content item in a local cloudlet will change over time as a result of requests, updates, arrivals as well as content distribution. On the basis of the available content state, the optimal problem will be periodically solved. Then data content will be re-distributed to cloudlets based on current optimal content distribution solution to a restart a new period.

3. Content State Transition

In the following, we would formulate a content distribution optimization problem based on a concept called content state, which is determined by content requests, editorial updates and new arrivals. The optimization objective of the problem is to minimize the average total content delivery latency for all users in a cooperative caching domain to obtain content service in DCSN. Data content can be distributed to local cloudlets according to the optimal solution to achieve the minimal average content delivery latency.

To derive the optimal content distribution, we first need to derive content state. Let us define two content states in C_n: V (valid) indicates validity and existence of a content item, while S (stale) indicates outdating or absence of an content item. Each cloudlet maintains a state list: $D = \{D_{i,n}\}_{\Omega\times N}$, where $D_{i,n} = \{V, S\}$ represents the state of O_i in C_n. Cloudlets in a cooperative caching domain share a global state list by periodically exchanging their content state to each other (*e.g.*, C_n notifies others about $D_{i,n}$) and updating the state list. Content state may be updated whenever a new content distribution, request, editorial update or new arrival occurs, as shown in **Fig. 4**. The state of an entry in the list changes from V to S when an editorial update or a new arrival occurs, while from S to V when a content request or a content distribution occurs. We detail the state transition in the following.

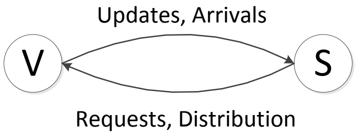


Fig. 4. State transition of content in state list

3.1 Content Request

When a request for O_i (i=1,2,…, Ω) (supposing $O_i \leftrightarrow M_{r,n}$) from a mobile user reaches C_n (n=1,2,…, N), there could be two cases:

1) *Hit*: $D_{i,n} = V$ obviously. Then C_n delivers O_i to the requesting user.
2) *Miss*: $D_{i,n} = S$, then C_n handles the miss by first searching in its state list: if $\exists m$ (m=1,2,…,N, $m \neq n$), $D_{i,m} = V$, then C_n fetches O_i from C_m (if there are multiple values of m, the non-overloaded cloudlet nearest to C_n is chosen). If $\forall m$ (m=1,2,…,N, $m \neq n$), $D_{i,m} = S$, C_n will forward the request to DCs, store the responded item and then send it to the requesting users. $D_{i,n}$ is then set to V.

Let us use an example to illustrate the content state transition due to content requests. In **Fig. 5**, the state transition takes place when content requests occur, where C_n has received the request of $M_{r,n}$ (r=2, 6, 12) from mobile users.

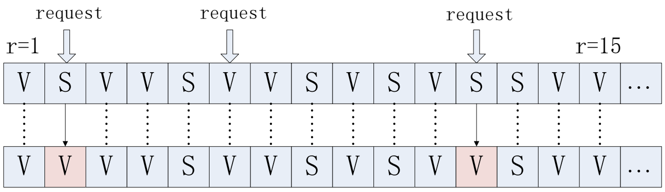

request
request
request
r=1
r=15
V S V V S V V S V S V S S V V …
V V V V S V V S V S V V S V V …

Fig. 5. State transition in state list when content is requested

3.2 Editorial Updates

CPs periodically update the content items in DCs. After updating a specific item (*e.g.*, a webpage) in DCs, the corresponding pre-allocated content item in cloudlets becomes invalid, and thus the states of the corresponding entries of the updated content in state list should be set to S. When a new request for the updated content arrives, the cloudlet will read corresponding content item from DCs, send it to the requesting user, and update the content in its cache. The example in **Fig. 6** shows the state transition when editorial updates occur, where CPs has updated $M_{r,n}$ (r=4, 11) in DCs.

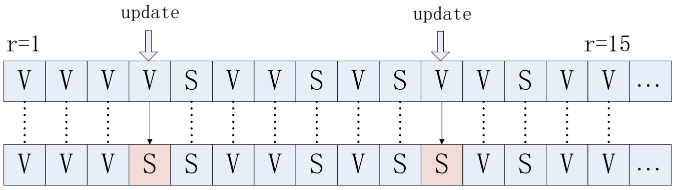

update
update
r=1
r=15
V V V V S V V S V S V V S V V …
V V V S S V V S V S S V S V V …

Fig. 6. State transition in state list when content is updated

3.3 New Arrivals

Besides updates, CPs may regularly inject new content items into DCs. When a new content item arrives, it does not exist in the cache of cloudlets in cooperative caching domain. Once being informed about new arrivals by DCs, the cloudlets will insert S into the corresponding entries in respective state list according to the popularity ranking in each CSA. When a newly arrived content item is requested by a user, the local cloudlet will fetch the content item from DCs and send it to the requesting user. The example in **Fig. 7** shows the change of content state when new arrivals occur, where the CP has injected new content $M_{r,n}$ (r=3, 9) in DCs.

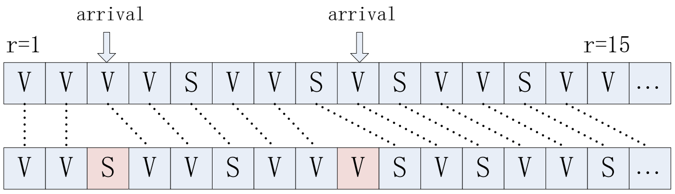
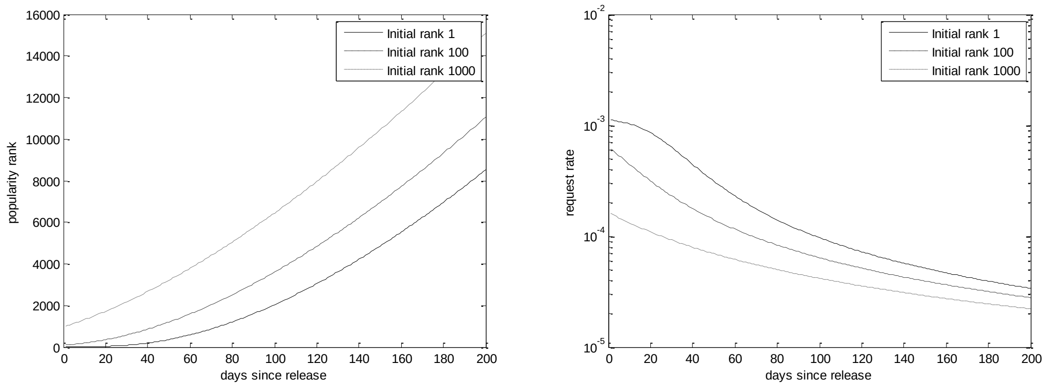

arrival
arrival
r=1
r=15
V V V V S V V S V S V V S V V …
V V S V V S V V V S V S V V S …

Fig. 7. State transition in state list when new content arrives

As shown in **Fig. 7**, new arrival means new item being inserted into the state list. This will cause “right shift” of content state: the state of the new content is inserted into the corresponding entries in state list based on the popularity ranking, while old entries are “exported” from higher ranking and “imported” to lower ranking. It means the popularity of a content item in a CSA will decrease over time, resulting in a decrease of request rate. **Fig. 8** shows popularity raking *vs.* time in an example of the Zipf-Mandelbrot distribution with $\beta = 0.8$ and $k = 20$. The details of the raking calculation will be given in the next section. Note that the last ranked content item will be purged if the cache is full and a new item is inserted.



(a). Popularity rank *vs.* time **(b).** Request rate *vs.* time

Fig. 8. Popularity characteristics decay

4. State Based Content Distribution (SCD)

In our proposed SCD framework, during off-peak hours, DCs pre-allocate data content in each cloudlet according to a content distribution matrix which will be determined by an optimal content distribution model. The purpose of pre-allocation is to minimize the average total content delivery latency for users to obtain required content in the network. The content distribution during off-peak hours is cheap or even “for free” as the data transmissions use “idle gap”, say night time.

Content distribution strategies to a great extent determine the content delivery latency. On the one hand, if at most only one copy for a content item is stored, the cached content item in all storage has the highest diversity so that the content can be fetched within a cooperative caching domain (between cloudlets) as much as possible. Thus the latency incurred by the content transmission from other cooperating cloudlets to users could be dominant due to increasing inter-cloudlet traffic. On the other hand, if multiple copies of popular content items are cached in more than one cloudlet in a cooperative caching domain, more requests can be locally met without using inter-cloudlet exchange. It is somehow contradictory to the requirement of content diversity, and thus the delivery latency from DCs to users could be dominant due to increasing outbound traffic. To optimize Quality of Experience (QoE) of all users, it is important to develop effective caching policy to minimize the total content delivery latency. In the following we formulate a state based content distribution model.

4.1 Optimal Content Distribution Model

First, we define a content distribution matrix $X = \{x_{i,n}\}_{\Omega \times N}$, where element $x_{i,n} = 1$ means that O_i is distributed to C_n, while $x_{i,n} = 0$ otherwise. Let $p_R(i,n)$, $p_U(i,n)$, $p_A(i,n)$ (abbreviated as p_R, p_U, p_A, respectively) denote the probabilities that O_i with rank *r* in C_n (*i.e.*, $O_i \leftrightarrow M_{r,n}$, without loss of generality) is subject to request, update and arrival, respectively. Then we have

$$p_R = 1 - [1 - f_R(i,n)]^R, \tag{1}$$

where $f_R(i,n)$ denotes the probability distribution of content requests. Note that $f_R(i,n)$ is the probability that O_i is requested in a single request. Then $1 - f_R(i,n)$ is the probability that O_i is not requested. $[1 - f_R(i,n)]^R$ is the probability that O_i is not requested in R attempts and $1 - [1 - f_R(i,n)]^R$ is the probability that O_i is requested at least once in R attempts. Similarly,

$$p_U = 1 - [1 - f_U(i,n)]^U, \tag{2}$$

$$p_A = 1 - [1 - f_A(i,n)]^A, \tag{3}$$

where $f_U(i,n)$ and $f_A(i,n)$ denote the probability distribution with respect to updates and arrivals respectively.

Let us first consider the case *r*=1 and let $P_{i,n}^V$ and $P_{i,n}^S$ denote the probabilities that $D_{i,n} = V$ and $D_{i,n} = S$ respectively. On the basis of the state transition depicted in **Fig. 4**, we have

$$P_{i,n}^V = x_{i,n} q_U q_A + (1 - x_{i,n}) p_R q_U q_A, \tag{4}$$

$$P_{i,n}^S = x_{i,n}(1 - q_U q_A) + (1 - x_{i,n})(1 - \ p_R q_U q_A), \tag{5}$$

where $q = 1 - p$.

Next let us consider the case *r*>1. Recall that new arrivals cause "right shift" in the state list: new entries are inserted and old entries are "exported" from higher ranking and "imported" to lower ranking. Furthermore, let us suppose a mapping: $O_j \leftrightarrow M_{k,n}$. Let $\rho_{k,n}^{V\rightarrow}$ and $\rho_{k,n}^{S\rightarrow}$ denote the probabilities that the exported entry from $M_{k,n}$ is V and S respectively, and then

$$\rho_{k,n}^{V\rightarrow} = x_{j,n} q_U + (1 - x_{j,n}) p_R q_U, \tag{6}$$

$$\rho_{k,n}^{S\rightarrow} = x_{j,n} p_U + (1 - x_{j,n})(1 - \ p_R q_U). \tag{7}$$

Assume that after "right shift", the entry imported to $M_{r,n}$ corresponds to the entry exported from $M_{k,n}$. It is obvious that *k*<*r* and the difference between *r* and *k* depends on the number of arrivals in the range from 1 to *r*-1. We can compute mean difference as $E = \sum_{i=1}^{r-1} A f_A(i,n)$. In more detail, k must be a positive integer and we may write $k \approx max\ (r - \lceil E \rceil, 1)$. Let $\rho_{r,n}^{V\leftarrow}$ and $\rho_{r,n}^{S\leftarrow}$ denote the probabilities that the imported entry to $M_{r,n}$ is V and S respectively, and then

$$\rho_{r,n}^{V\leftarrow} \approx \rho_{max(r-\lceil E \rceil,1),n}^{V\rightarrow} \tag{8}$$

$$\rho_{r,n}^{S\leftarrow} \approx \rho_{max(r-\lceil E \rceil,1),n}^{S\rightarrow}. \tag{9}$$

As discussed in 3.3, the state of the inserted entries for new arrivals is S, while the state of other entries stays the same with which they are exported from. Thus we have

$$P_{i,n}^{V} = q_A \rho_{r,n}^{V\leftarrow}, \tag{10}$$

$$P_{i,n}^{S} = q_A \rho_{r,n}^{S\leftarrow} + p_A. \tag{11}$$

Combining (6) (8) (10) and (7) (9)(11) yields

$$P_{i,n}^{V} = q_A\left[x_{j,n} q_U + \left(1 - x_{j,n}\right) p_R q_U \ \right], \tag{12}$$

$$P_{i,n}^{S} = q_A\left[x_{j,n} p_U + (1 - x_{j,n})(1 - \ p_R q_U)\right] + p_A. \tag{13}$$

For *r*=1, it is obvious that the mean number of new arrivals before $M_{1,n}$ is zero (*i.e.* $E = 0$), and hence *k*=1 and *j*=*i*. Manipulating (4) (5) and (12) (13), we can obtain unified expressions for *r*=1,2,…, *Ω* as follows,

$$P_{i,n}^{V} = q_R q_U q_A \cdot x_{j,n} + p_R q_U q_A, \tag{14}$$

$$P_{i,n}^{S} = -q_R q_U q_A \cdot x_{j,n} + 1 - p_R q_U q_A. \tag{15}$$

Consider the following three cases: *1)* If O_i has already been cached in C_n, the latency for a user to obtain O_i from C_n directly is t_1; *2)* If O_i is not in C_n but has been distributed to at least one other cloudlet in the cooperative caching domain, the latency for a user to obtain O_i from that cloudlet is t_2; *3)* If O_i is not cached in any cloudlet in the cooperative caching domain, the latency for a user to obtain O_i from DCs is t_3. We have

$$\begin{cases} t_1 = t_{local} \\ t_2 = t_{local} + t_{inter} \\ t_3 = t_{local} + t_{outer} \end{cases} \tag{16}$$

where t_{local} is the latency for a user to obtain content from a local cloudlet. t_{outer} is the latency for a local cloudlet to obtain content from DCs. Generally t_{local} and t_{outer} can be regarded as constant. t_{inter} is the latency of a content item transferred between two cloudlets in the cooperative caching domain, generally proportional to the distance between the two cloudlets, in terms of hop count. Let $d_{n,m}$ denotes the distance between C_n and C_m, then we have

$$t_{inter} = \alpha \cdot d_{n,m}, \tag{17}$$

where α is a constant.

A content item can be cached in a number of different cloudlets. The latency to obtain a specific content item from different cloudlets varies since the distances between users and these cloudlets are different. If O_i requested by users in C_n's CSA has not been cached in C_n, but has been distributed to multiple other cloudlets in the cooperative caching domain, C_n will choose O_i in the cloudlet which is the closest to C_n. Thus we can re-write (17) as

$$t_{inter} = \alpha \cdot min\left\{d_{n,m},\ x_{i,m} = 1\, for\ m = 1\ to\ N, m \neq n\right\}. \tag{18}$$

Let $t_{i,n}$ denote the average latency for a user to get O_i when he sends request to C_n, which can be expressed as

$$t_{i,n} = P_{i,n}^{V} \cdot t_1 + P_{i,n}^{S}\left(1 - \prod_{\substack{m=1 \\ m\neq n}}^{N} P_{i,m}^{S}\right) \cdot t_2 + \prod_{m=1}^{N} P_{i,m}^{S} \cdot t_3, \tag{19}$$

Let $H = \{H_1, H_2, \dots, H_N\}$ respectively denote the cache space of $C = \{C_1, C_2, \dots, C_N\}$, and $S = \{s_1, s_2, \dots, s_\Omega\}$ denotes the sizes of *Ω* content items in the system. The space occupied by content items in C_n must not exceed the total cache space of C_n, thus we have

$$\sum_{i=1}^{\Omega} x_{i,n} \cdot s_i \leq H_n. \tag{20}$$

Similarly, the transmission capacity of cloudlets is also limited. Thus a limited number of users can be served simultaneously. Let $B = \{B_1, B_2, \ldots, B_N\}$ denote the transmission bandwidth of $C = \{C_1, C_2, \ldots, C_N\}$, and $R = \{r_1, r_2, \ldots, r_\Omega\}$ denote the downlink rates to transmit content. When C_n provides content service to local users, the following constraint should be met

$$\sum_{i=1}^{\Omega} f_R(i,n) \cdot x_{i,n} \cdot r_i \leq B_n. \tag{21}$$

Based on the discussions above, we can formulate a state based content distribution model, with objective of minimizing the average total content delivery latency for all users in a cooperative caching domain to obtain content service in DCSN as follows.

$$\begin{aligned}
(P) \quad & min \quad \sum_{n=1}^{N} \sum_{i=1}^{\Omega} f_R(i,n) \cdot t_{i,n} \\
s.t. \quad & t_{i,n} = P_{i,n}^{V} \cdot t_1 + P_{i,n}^{S} \left(1 - \prod_{\substack{m=1 \\ m \neq n}}^{N} P_{i,m}^{S} \right) \cdot t_2 + \prod_{m=1}^{N} P_{i,m}^{S} \cdot t_3, \\
& \qquad \forall\, n \in \{1, \ldots, N\}, \forall\, i \in \{1, \ldots, \Omega\} \\
& \sum_{i=1}^{\Omega} x_{i,n} \cdot s_i \leq H_n, \qquad \forall\, n \in \{1, \ldots, N\} \\
& \sum_{i=1}^{\Omega} f_R(i,n) \cdot x_{i,n} \cdot r_i \leq B_n, \qquad \forall\, n \in \{1, \ldots, N\} \\
& x_{i,n} = 0,1, \qquad \forall\, n \in \{1, \ldots, N\}, \forall\, i \in \{1, \ldots, \Omega\}
\end{aligned}$$

This optimization is nonlinear integer 0-1 programing problem and is NP-hard. To prove this, let us examine a special case. Let there be K copies of O_i allocated in the cooperative domain. We are to minimize the average total content delivery latency in (P) so that the K cloudlets in which O_i is cached are required to be as close as possible to the rest $N - K$ cloudlets, since otherwise the inter-cloudlet average latency can be further reduced. Hence a part of the problem can be converted to a K-Center problem [21]: find K points in a given undirected complete weighted graph $G = (V, E)$, where $d(V_i, V_j)$ is the distance between a pair of fixed points (V_i, V_j) $(1 \leq i, j \leq N)$ and $d\left(V_i, V_j\right) > 0$, to minimize the maximum value of the minimum distance between the K points and other $N - K$ points. K-Center problem is known to be NP-hard and the algorithm complexity is $O(k(n-k)^2)$ [21]. Thus the optimization problem (P) is also NP-hard. In this paper, we employ software tools, such as Lingo [22], to numerically solve the problem.

4.2 Algorithm for Content Distribution

To present the idea of SCD more clearly, we describe the algorithm of the optimal content distribution in Table 1. In a cooperative caching domain, each cloudlet shares and maintains a global state list $D = \{D_{i,n}\}_{\Omega \times N}$. These cloudlets periodically collect information about updates and arrivals from DCs and change content state in state list corresponding to the information. Then the optimal solution of the 0-1 variables in content distribution matrix $X = \{x_{i,n}\}_{\Omega \times N}$ can be determined by solving the problem (P) based on current content state. During off peak hours, content items can be delivered from DCs to cloudlets on the basis of the optimal solution: If $x_{i,n} = 1$, O_i is distributed to be cached in C_n, while $x_{i,n} = 0$ otherwise. The cloudlets provide content service for requesting users whenever receiving a request for data content, and the state of a certain content item will change over time as a result of content request. The algorithm will be repeated by starting a new content distribution period. The final state in the prior period will then be the initial sate on the following period.

Table 1. Algorithm for content distribution in cloudlets

```
SCD for $C_n$ ($n$=1,2,...$N$) on day $t$.
1) Get initial state list $D = \{D_{i,n}\}_{\Omega \times N}$ from day $t$-1.
2) Collect information about updates and arrivals from DCs.
2) Calculate the optimal distribution matrix $X = \{x_{i,n}\}_{\Omega \times N}$ by solving ($P$).
3) Pre-allocate content during off peak hours:
    For $i$=1 to $\Omega$
      For $n=1$ to N {
        If ($x_{i,n}$ == 1 && $D_{i,n}$ == $S$)
          Cache $O_i$ in $C_n$, set $D_{i,n} = V$;
        Else if ($x_{i,n}$ == 0 && $D_{i,n}$ == $V$)
          Delete $O_i$ in $C_n$'s cache, set $D_{i,n} = S$;
      }
4) Provide content service for requesting users.
5) Go to (1) to restart a new period. The final state on day t will then be the initial
    sate on day $t$+1.
```

In a practical communication system, cloud service providers deploy a plurality of cloudlets, which are deployed and co-located with base stations in mobile cellular networks. To implement the proposed cooperative caching strategy, cloudlets have to exchange information with each other (*i.e.* information about cached content, shared data content) and with data centers (*i.e.* type of content, new arrivals, updates, fetched content *etc.*) by exploiting the communications between the traditional base stations, *e.g.*, using the X2 interface in LTE technique to transmit data between NodeBs [25]. With the information exchanged, the proposed content distribution optimization problem can be solved in cloudlets. Then the content distribution matrix can be determined and according to the optimal solution, the cloudlets fetch the corresponding content items from data centers and cache them in local caches.

We would like to mention that our proposed model is not limited to a specific content popularity distribution, network topology, content update and caching strategy. Specifically, no matter which caching strategy is adopted and what distribution the content requests, updates or arrivals are subject to, the optimal content distribution solution can be determined by deriving the probabilities of content status ($P_{i,n}^V$ and $P_{i,n}^S$ or likewise) on the basis of corresponding strategy and solving the optimization model.

5. Performance Evaluation

In this section, we examine the performance of our proposed SCD. Computer simulation experiments are conducted to compare the performance of SCD with a number of known content caching schemes, including Least Recently Used (LRU) [13], Least Frequently Used (LFU) [14], and Randomized Replacement (RR) [16]. LRU is a cached content replacement strategy which exploits the temporal locality seen in the past request streams to predict future accesses to content, and removes the least recent used content. LFU is also a cached content update strategy which uses popularity for future decisions and removes the least frequent used content. RR uses randomized decisions to find a content item for replacement, aiming to reduce the complexity of the replacement process without compromising the quality too much. These three caching strategies are known and widely used as comparison references [7][23].

We describe the optimization model derived in Section 3.2 in Lingo [22] to numerically solve the problem described in Section 4. Lingo is software which has been developed to solve linear and nonlinear optimization problems, and is embedded with a kind of optimization modeling language exploited to express large-scale problems. The highly efficient solver of Lingo provides high-speed problem solution and analysis of results. The optimal content distribution solution is determined by solving the problem in Lingo. Then we carry out computer-based simulation of the content caching and distribution process in Microsoft Visual C++ by using the solution of Lingo and the performance in terms of cache hit rate, content delivery latency and traffic volume is evaluated.

5.1 Performance Metrics

We adopt three metrics, namely cache hit rate (HR), average total content delivery latency (ATL) and data traffic volume (DTV) (including local traffic, inter-cloudlet traffic and outbound traffic), which are defined below. Furthermore, we define a factor called cache redundancy (CR), to characterize the caching capability of a cooperative caching domain.

A. Cache Hit Rate (HR)

A cache stores distributed content and handles content requests by either of the following two cases: *1)* If the requested content is stored, sending it to the requesting user. This is referred to as a cache hit. *2)* If the content is not stored, forwarding the request to a different provider, *i.e.*, other cloudlets or DCs. This is referred as a cache miss. Then the cache HR may be defined as the fraction of requests that results in hits:

$$HR \overset{\text{def}}{=} \frac{R-M}{R} = 1 - \frac{M}{R},$$

where R is the number of requests a cloudlet receives per time unit (say a day), and M is the number of cache misses. We can readily obtain the number of cache misses M as the expected number of request driven transitions from S to V,

$$M = \sum_{i=1}^{\Omega} P_{i,n}^{S} \cdot p_R(i,n).$$

B. Average Total Content Delivery Latency (ATL)

The ATL for users in a cooperative caching domain to obtain content service in DCSN is also an important metric for user QoE. ATL is defined as

$$ATL = \sum_{n=1}^{N} \sum_{i=1}^{\Omega} f_R(i,n) \cdot t_{i,n}.$$

Note that ATL is also our optimization objective in (*P*). Lower ATL means the mobile users can receive better content service.

C. Data Traffic Volume (DTV)

Let V_1, V_2 and V_3 denote the local traffic volume, inter-cloudlet traffic volume and outbound traffic volume, respectively. We calculate them by adding up the total amount of content transferred from local cloudlets, other cloudlets and DCs, respectively, to users. A content item is delivered to users from a local cloudlet directly if it is locally cached. So V_1 can be expressed as

$$V_1 = \sum_{n=1}^{N} \sum_{i=1}^{\Omega} f_R(i,n) \cdot s_i \cdot x_{i,n},$$

where s_i denotes the size of O_i. Similarly, V_2 and V_3 can be respectively obtained.

D. Cache Redundancy (CR)

We can write the total cache space of all cloudlets in a cooperative caching domain

$$H_{total} = \sum_{n=1}^{N} H_n,$$

and the total size of the Ω content items in the system is expressed as

$$S_{total} = \sum_{i=1}^{\Omega} s_i.$$

To characterize the relevant caching capability of a cooperative caching domain, we define the ratio of H_{total} and S_{total} as a factor called cache redundancy:

$$\eta \stackrel{\text{def}}{=} \frac{H_{total}}{S_{total}}.$$

η is a metric value characterizing the redundancy of content in a cooperative caching domain. The greater η represents a larger degree of content redundancy in the domain, which means more replicas can be cached in multiple cloudlets and thus the average content delivery latency will be lower. In addition, the cache space of the cloud system will be relatively richer if η is greater, so more flexible caching strategy can be adopted. There are following two cases:

1) $\eta \leq 1$, which means $H_{total} \leq S_{total}$. Some content may not be cached in cloudlets.
2) $\eta > 1$, which means $H_{total} > S_{total}$. All content can be cached and some may be duplicated in multiple cloudlets to minimize the content delivery latency.

Different kinds of applications will significantly affect the performance of the proposed scheme. Let us consider the following two scenarios. (1). Web objects are dominating in mobile Internet applications. The objects cached in cloudlets are a large amount of data content with small sizes, which may face frequent editorial updates in data centers. Thus frequent information exchange between data centers and cloudlets is needed to update the cached object in time. The cost of frequent communications between data centers and cloudlets should be taken into account, which may affect the overall performance of the proposed scheme to a certain extent. (2). Video streams are dominating in mobile Internet applications, which are typically large sized and rarely updated. Then we should consider more about the impact of latency on overall performance to transmit large-size files in inter-cloudlet traffic if cooperative caching strategy is adopted in DCSN. Furthermore, the deployment of cloudlets can also affect the performance of the proposed scheme. On the one hand, if a large number of cloudlets are deployed in a cooperative caching domain, the cooperation among these cloudlets can play an important role, and the performance can be improved by mutual sharing data content among cloudlets. On the other hand, if only a small number of cloudlets are deployed in a cooperative caching domain, the cooperative caching strategy do not benefit so much that we may consider using existing caching strategies which are applied to single cache with lower overheads, such as LFU, LRU or RR.

5.2 Numerical Results and Discussions

In the simulation experiments, we consider a scenario where video content distribution service is dominant in mobile Internet applications. The authors of [20] observe that circular curves of file popularity can be captured by Zipf – Mandelbrot law, reflecting a set of characteristics critical for streaming media services. Let us assume there are totally $\Omega = 16{,}000$ video tracks with relative popularities following the Zipf-Mandelbrot distribution with $\beta = 0.75$ and $k = 80$ similar to that in [20]. The video frame size distribution is subject to an exponential distribution with mean of 200 according to the analysis of video frame size distribution, similar to that in [24]. Moreover, on average $U = 320$ video tracks are updated per day and the popularity of an updated video can be modelled as a sample from a modified Zipf distribution with $\beta = 0.075$. Finally we assume that 160 new video tracks are added per day and the initial popularity of a new track can be modelled as a sample from the same Zipf-Mandelbrot distribution. As for the domain model, we consider $N = 30$ cloudlets in a domain with $R = 100$ requests on average per cloudlet. The cloudlets in a domain are supposed to have unified cache size. The delivery latency and bandwidth in access network and core network are determined according to 3GPP LTE standard [25], which are reasonable for the next generation optical-wireless converged networks.

Fig. 9 depicts the cache HR as a function of cache redundancy η. η ranges from 0 to 30, where $\eta = 0$ represents no caching is applied. We can observe that cache HR of all schemes increases with η. As expected, the cache HR of SCD performs the best. For example, cache HR of SCD is higher than LFU, LRU and RR by around 15%, 27%, and 76%, respectively, when η is 15. The difference is small when η is small (close to 0) and large (close to 30) and maximum when η is median, say around 12. This is because that when η is close to 0, a cloudlet has little cache space and only small amount of data is cached. In this case, only very little data can be cached and the mentioned caching strategies play little role. With a high probability a local cloudlet has to fetch a content item from DCs when receiving a user request no matter which strategy is adopted. As a result, both SCD and LFU/LRU schemes achieve poor performance in terms of hit rates, outbound traffic volume and content delivery latency. On the other hand, when η is very high, the cache space of a cloudlet is large enough to cache all the content in DCs. In this case, most user requests can be locally met by using any caching strategies, and thus both SCD and LFU/LRU schemes can achieve good performance.

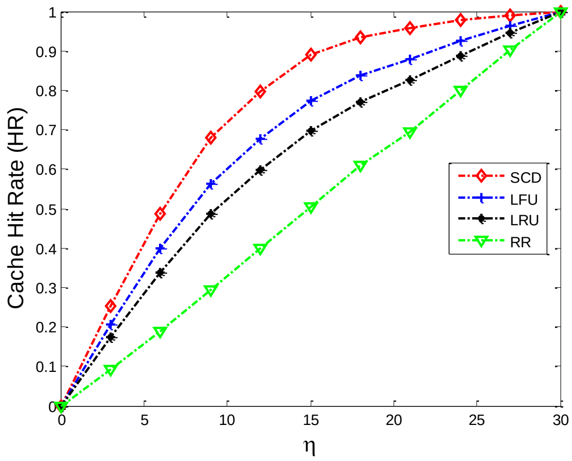


Fig. 9. HR *vs.* η for SCD, LRU, LFU and RR respectively

Fig. 10 depicts the ATL, which is the optimization objective of problem (P), as a function of cache redundancy η for four caching strategies. We can observe that the ATL of all schemes decreases with η. Lager cache space results in a lower ATL, because more content can be locally cached thus less time is needed to fetch a specific content item from local cloudlets than from DCs. The result is as expected that SCD has the lowest ATL. For example, the ATL of SCD is lower than LFU, LRU and RR by around 14%, 26%, and 72%, respectively, when η is 15. The difference of ATL behaves similarly to that of cache HR, as mentioned before.

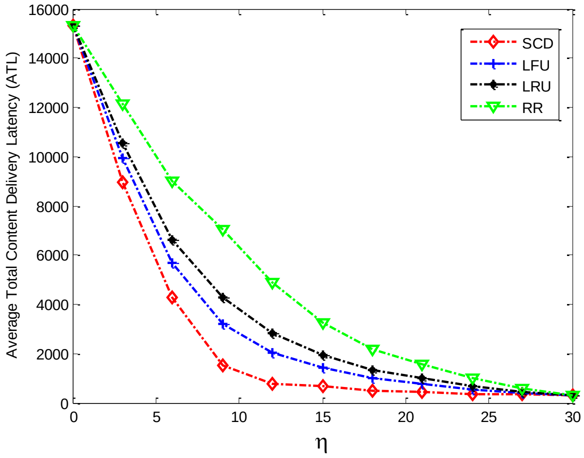
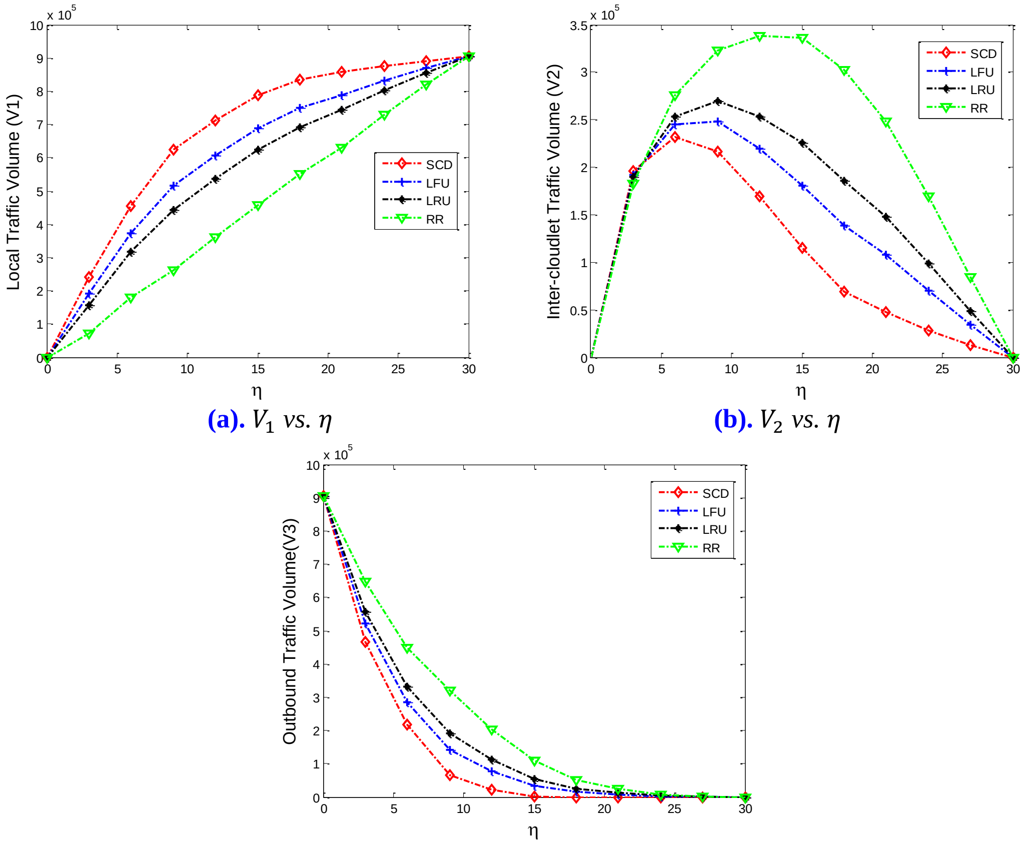


Fig. 10. ALT *vs.* η for SCD, LRU, LFU and RR respectively

Figs. 11 depicts the DTV, including local traffic volume V_1, inter-cloudlet traffic volume V_2 and outbound traffic volume V_3 as a function of cache redundancy η. We can observe that V_1 of all schemes increases with η while the tendency of V_3 is on the contrary. V_2 is small when η is small (close to 0) and large (close to 30) and maximum when η is median. In one extreme case, all users have to fetch content from DCs if there is no local cache ($\eta = 0$), where there is only outbound traffic. In another extreme case where there is fully sufficient cache space ($\eta = 30$), all content can be fetched from local cloudlets, which results in only local traffic.



(a). V_1 *vs.* η

(b). V_2 *vs.* η

(c). V_3 *vs.* η

Fig. 11. DTV *vs.* η for SCD, LRU, LFU and RR respectively

Simulation results show that cache performance can be significantly improved by using our proposed SCD. Specifically, SCD can effectively improve cache HR, reduce outbound traffic volume and content delivery latency in comparison with LRU, LFU and RR. The benefits of SCD mainly come from the cooperation among multiple cloudlets in a cooperative caching domain and using our optimization model. Moreover, we take into account the updates and arrivals, which meets the dynamic needs of users in our optimization model. The cloudlets co-located with base stations which are deployed far away might cause relative high latency in practical communication system. However, the performance of content delivery latency can be significantly improved by local cooperative caching especially when η is median. On the other hand, when η is very low/high, most user requests can be locally/remotely met by local cloudlets/DCs, and thus there will be little inter-cloudlet traffic. In other words, the overall improvement on data delivery performance brought by local and cooperative caching is significantly greater than the extra delay which could be incurred in the data transfer between cloudlets.

However, we should note that SCD may not be suitable for the case where the cache performance is not that obviously improved when η is very low/high, due to its higher complexity compared with LFU/LRU. As illustrated in the last paragraph of Section 4.1, a part of the problem can be converted to a K-Center problem, which is known to be NP-hard and the algorithm complexity is $O(k(n-k)^2)$ [21]. As for LFU/LRU, which uses recency/frequency respectively as a main factor, there are K least recently/frequently used objects should be removed from the N cloudlets if K copies of O_i are distributed in the cooperative domain. The problem can be converted as finding the smallest K numbers among the last recency/frequency ranked numbers in N cloudlets. The algorithm complexity is $O(k(n-k))$ using a quick sorting algorithm. It is evident that the proposed scheme has a higher complexity than LFU/LRU, thus we should make a trade-off when adopting SCD in different cases, especially for the cases where the performance improvement is less obvious when η is very low/high.

6. Conclusion

In this paper, we have constructed a cooperative caching architecture where several local caches of distributed cloudlets in DCSN work cooperatively. We have proposed a State based Content Distribution strategy (SCD) in DCSN for future mobile communication networks. Considering the layered feature of DCSN, multimedia services can be pre-allocated in local cloudlets which are distributed and equipped with powerful storage, to avoid repeated, redundant content transmissions in the network. Content requests, editorial updates and new arrivals are considered to determine the transition of content state. To minimize the content delivery latency for mobile users, we formulate a state based content distribution optimization model. Optimal content distribution can be determined by solving the model. We have carried out computer simulations to verify the effectiveness of our proposed cooperative caching architecture. The numerical results indicate that the performance of content distribution benefits from our proposed cooperative content caching and distribution strategy in mobile networks. Our proposed SCD outperforms LRU, LFU and RR, in terms of cache hit rate, content delivery latency and traffic volume. The performance improvement via content caching and distribution would enhance the energy efficiency of the next generation networks and address the increasing demand for mobile multimedia and data services in emerging 5G systems.

References

[1] http://www.cisco.com/web/CN/aboutcisco/news_info/corporate_news/2013/06_27.html.

[2] M. Felemban, S. Basalamah, and A. Ghafoor, “A Distributed Cloud Architecture for Mobile Multimedia Services,” *IEEE Network*, October, 2013. Article (CrossRef Link).

[3] J. Erman, A. Gerber, M. Hajiaghayi, *et al.*, “to cache or not to cache the 3G case,” *IEEE Internet Computing*, vol. 15, no. 2, pp. 27–34, Mar. 2011. Article (CrossRef Link).

[4] S. Woo, E. Jeong, S. Park, *et al.*, “Comparison of Caching Strategies in Modern Cellular Backhaul Networks,” *ACM MobiSys*, June, 2013. Article (CrossRef Link).

[5] N. Abedini and S. Shakkottai, “Content caching and scheduling in wireless networks with elastic and inelastic traffic,” *IEEE transaction on networking*, 22(3), 2014. Article (CrossRef Link).

[6] H. Ahlehagh and S. Dey, “Video caching in radio access network: impact on delay and capacity,” *IEEE WCNC*, 2012. Article (CrossRef Link).

[7] X. Wang, M. Chen, T. Taleb, *et al.*, “Cache in the air: exploiting content caching and delivery techniques for 5g systems,” *IEEE Communications Magazine* 52.2 (2014): 131-139. Article (CrossRef Link).

[8] B. A. Ramanan, L. M. Drabeck, M. Haner, *et al.*, “Cacheability Analysis of HTTP traffic in an Operational LTE Network,” *Wireless Telecommun*. Symp., Apr. 2013. Article (CrossRef Link).

[9] A. Arvidsson, A. Mihaly and L. Westberg, “Optimized local caching in cellular mobile network,” *Computer Networks*, vol.55, 2011. Article (CrossRef Link).

[10] M. Satyanarayanan, P. Bahl, R. Caceres, *et al.*, “The case for VM-based cloudlets in mobile computing,” *IEEE Pervasive Computing*, vol. 8, no. 4, pp. 14–23, Oct. 2009. Article (CrossRef Link).

[11] W. Qing, H. Zheng, W. Ming, *et al.*, “CACTSE: Cloudlet Aided Cooperative Terminals Service Environment for Mobile Proximity Content Delivery,” *China Communications*, Vol.10, June, 2013. Article (CrossRef Link).

[12] D. Fesehaye, Y. Gao, K. Nahrstedt, *et al.*, “Impact of Cloudlets on Interactive Mobile Cloud Applications,” 16th *International Enterprise Distributed Object Computing Conference*, 2012. Article (CrossRef Link).

[13] M. Abrams, C. R. Standridge, G. Abdulla, *et al.*, “Caching proxies: Limitations and potentials,” WWW-4, *Boston Conference*, 1995. Article (CrossRef Link).

[14] M. Arlitt, L. Cherkasova, J. Dilley, *et al.*, “Evaluating content management techniques for web proxy caches,” *ACM SIGMETRICS Performance Evaluation Review*, vol. 27, no. 4, pp. 3−11, 2000. Article (CrossRef Link).

[15] M. Arlitt, R. Friedrich, and T. Jin, “Workload characterization of a web proxy in a cable modem environment,” *ACM SIGMETRICS Performance Evaluation Review*, vol. 27, no. 2, pp. 25−36, 1999. Article (CrossRef Link).

[16] K. Psounis and B. Prabhakar, “A randomized web-cache replacement scheme,” in *Proc. of IEEE Infocom* 2001, vol. 3, pp. 1407−1415, 2001. Article (CrossRef Link).

[17] Y. Bernet, “The complementary roles of RSVP and differentiated services in the full-service QoS network,” *Communications Magazine*, *IEEE*38.2 (2000): 154-162. Article (CrossRef Link).

[18] S. Glassman, “A caching relay for the World Wide Web,” *Computer Networks and ISDN Systems* 27.2: 165-173, 1994. Article (CrossRef Link).

[19] H. Yu, D. Zheng, B. Y. Zhao, *et al.*, “Understanding user behavior in large-scale video-on-demand systems,” *ACM SIGOPS Operating Systems Review*, Vol. 40. No. 4. ACM, 2006. Article (CrossRef Link).

[20] W. Tang, Y. Fu, L Cherkasova, *et al.*, “Modeling and generating realistic streaming media server workloads,” *Computer Networks* 51.1: 336-356, 2007. Article (CrossRef Link).

[21] M. R. Garey and D. S. Johnson, “Computers and intractability: a guide to the theory of NP-completeness,” 1979[J]. San Francisco, LA: Freeman, 1979.

[22] Extended LINGO Release 8.0 HLP. LINDO Systems Ins. 2003.

[23] J. Gu, W. Wang, A. Huang, *et al.*, “Distributed cache replacement for caching-enable base stations in cellular networks,” *Communications (ICC)*, 2014 *IEEE* International Conference on. *IEEE*, 2014. Article (CrossRef Link).

[24] D. M. B. Masi, M. J. Fischer, D. A. Garbin, “Video Frame Size Distribution Analysis,” *The Telecommunications Review* 2008, Volume 19, Sept 2008.

[25] J. Shen, S. Suo, H. Quan, *et al.*, 3GPP long term evolution: principle and system design[J]. 2008.

[26] N. D. Han, Y. Chung, M. Jo, “Green data centers for cloud-assisted mobile ad hoc networks in 5G,” *IEEE Network*, 29.2: 70-76, 2015. Article (CrossRef Link).

[27] B. Liu，V. Firoiu，J. Kurose, *et al.*, “Capacity of Cache Enabled Content Distribution Wireless Ad Hoc Networks,” *Mobile Ad Hoc and Sensor Systems (MASS)*, 2014 *IEEE* 11th *International Conference on. IEEE*, 2014: 309-317, 2014. Article (CrossRef Link).

[28] I. Psaras, W. K. Chai, G. Pavlou, “In-Network Cache Management and Resource Allocation for Information-Centric Networks,” *IEEE Transactions on Parallel & Distributed Systems* 25.11:2920 – 2931, 2014. Article (CrossRef Link).

[29] L. Al-Kanj, Z. Dawy, E. Yaacoub, “Energy-Aware Cooperative Content Distribution over Wireless Networks: Design Alternatives and Implementation Aspects,” *IEEE Communications Surveys & Tutorials* 15.4(2013):1736 - 1760. Article (CrossRef Link).

[30] X. Zheng, C. Cho, Y. Xia, “Algorithms and Stability Analysis for Content Distribution over Multiple Multicast Trees,” in *Proc. of the IEEE Conference on Decision & Control* (2014):1. Article (CrossRef Link).

[31] A. Damnjanovic, J. Montojo, Y. Wei, *et al.*, “A survey on 3GPP heterogeneous networks,” *IEEE Wireless Communications*, 2011, 18(3): 10-21. Article (CrossRef Link).

[32] C. Ranaweera, E. Wong, C. Lim, *et al.*, “Next generation optical-wireless converged network architectures,” *IEEE Network*, 26(2): 22-27, 2012. Article (CrossRef Link).

[33] "White Paper on 5G Concept," *IMT-2020 (5G) Promotion Group*, Fb, 2015. Article (CrossRef Link).

[34] "Deliverable D6.4: Final report on architecture," *Mobile and wireless communications Enablers for the Twenty-twenty Information Society (METIS)*, Document Number: ICT-317669-METIS/D6.4. Article (CrossRef Link).

[35] "NGMN 5G Initiative White Paper," *NGMN Alliance*, Dec. 2014. Article (CrossRef Link).

[36] C. Liu, K. Sundaresan, M. L. Jiang, "The Case for Re-configurable Backhaul in Cloud-RAN based Small Cell Networks," in *Proc. of IEEE INFOCOM*. Article (CrossRef Link).

[37] S. Namba, T. Warabino, S. Kaneko, "BBU-RRH Switching Schemes for Centralized RAN," 7th *International ICST Conference on CHINACOM*, 2012. Article (CrossRef Link).

[38] M. Jo, T. Maksymyuk, B. Strykhalyuk, *et al.*, "Device-to-Device Based Heterogeneous Radio Access Network Architecture for Mobile Cloud Computing," *IEEE Wireless Communications*, Vol.12, No.3, June 2015. Article (CrossRef Link).

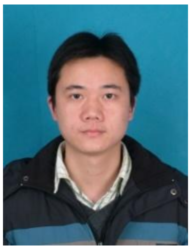
[39] *CMCC (China Mobile Communications Corporation)* white paper: "C-RAN the road towards green RAN," Version 2.5, Oct. 2011. Article (CrossRef Link).

[40] C.-L. I, C. Rowell, S. Han, *et al*., "Towards green and soft: A 5G perspective," *IEEE Commun*. Mag., vol. 52, no. 2, pp. 6673, Feb. 2014. Article (CrossRef Link).

[41] C.-L. I, C. Cui, J. Huang, *et al.*, "C-RAN: Towards open, green and soft RAN," *IEEE Netw.*, to be published.

Lirong Jiang received the B.S. degree in Electronic Information Science and Technology from University of Electronic Science and Technology of China (UESTC), Chengdu, in 2013. She is now a postgraduate student with the National Key Laboratory of Science and Technology on Communications in UESTC. Her research interest includes content distribution networks, caching in mobile networks and the security of communication in wireless sensor networks.

Dr. Gang Feng (M'01, SM'06) received his BEng. and MEng degrees in Electronic Engineering from the University of Electronic Science and Technology of China (UESTC), in 1986 and 1989, respectively, and the Ph.D. degrees in Information Engineering from The Chinese University of Hong Kong in 1998. He joined the School of Electric and Electronic Engineering, Nanyang Technological University in December 2000 as an assistant professor and was promoted as an associate professor in October 2005. At present he is a professor with the National Laboratory of Communications, University of Electronic Science and Technology of China. Dr. Feng has extensive research experience and has published widely in computer networking and wireless networking research. His research interests include resource management in wireless networks, next generation cellular networks, *etc.* Dr. Feng is a senior member of IEEE.



Shuang Qin received the B.S. degree in Electronic Information Science and Technology, and the Ph.D degree in Communication and Information System from University of Electronic Science and Technology of China (UESTC), in 2006 and 2012, respectively. He is currently a associate professor with National Key Laboratory of Science and Technology on Communications in UESTC. His research interests include cooperative communication in wireless networks, data transmission in opportunistic networks and green communication in heterogeneous networks.