**EMERGING TECHNOLOGIES & APPLICATIONS**

**Energy-Efficient Joint Caching and Transcoding for HTTP Adaptive Streaming in 5G Networks with Mobile Edge Computing**

**Renchao Xie1,2,\*, Zishu Li1,4, Jun Wu3, Qingmin Jia1, Tao Huang1,2**

1. State Key Laboratory of Network and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China
2. Purple Mountain Laboratories, Nanjing 211111, China
3. Dhurakij Pundit University, 110/1-4 Prachachuen Rd., Laksi, Bangkok 10210 Thailand
4. Institute of Sensing Technology and Business, BUPT, 214315, China
   * The corresponding author, email: Renchao\_xie@bupt.edu.cn

**Abstract:** With the new promising techniqueof mobile edge computing (MEC) emerging, by utilizing the edge computing and cloud computing capabilities to realize the HTTP adaptive video streaming transmission in MEC-based 5G networks has been widely studied. Although many works have been done, most of the existing works focus on the issues of network resource utilization or the quality of experience (QoE) promotion, while the energy efficiency is largely ignored. In this paper, different from previous works, in order to realize the energy efficiency for video transmission in MEC-enhanced 5G networks, we propose a joint caching and transcoding schedule strategy for HTTP adaptive video streaming transmission by taking the caching and transcoding into consideration. We formu-late the problem of energy-efficient joint cach-ing and transcoding as an integer program-ming problem to minimize the system energy consumption. Due to solving the optimization problem brings huge computation complexity, therefore, to make the optimization problem tractable, a heuristic algorithm based on sim-ulated annealing algorithm is proposed to iteratively reach the global optimum solution

with a lower complexity and higher accura-cy. Finally, numerical simulation results are illustrated to demonstrated that our proposed scheme brings an excellent performance.

**Keywords:** mobile edge computing; HTTPadaptive streaming; caching; transcoding; en-ergy efficiency

1. **INTRODUCTION**

With the emerging application of 4K/8K video, virtual reality (VR)/augmented reality (AR), and Internet of Things (IoT), the mobile data traffic is growing dramatically. Accord-ing to a recent report from Cisco [1], mobile data traffic will increase 7-fold between 2017 and 2022, up to 77.5EB per month by 2022, and nearly 82% of the all traffic is expected to be video traffic. An explosive growth of video traffic brings huge challenges for mobile network during video transmission, which requires that the network has the ability to provide higher data rate and lower latency for video transmission in a time varying wireless environment in order to satisfy the users’ qual-ity of experience (QoE). In order to cope with these challenges, the HTTP adaptive streaming

Received: Jan. 15, 2019

Accepted: Apr. 09, 2019

Editor: Yong Cui

|  |  |
| --- | --- |
| China Communications • July 2019 | 229 |

Authorized licensed use limited to: University of New South Wales. Downloaded on November 20,2020 at 02:28:37 UTC from IEEE Xplore. Restrictions apply.

In this paper, we have proposed an energy efficient joint cach-ing and transcoding schedule strategy to dynamically deter-mine the placement of requested video segments and the way of responding users’ content requests.

(HAS) technique has been designed to realize video transmission over mobile networks [2]-

1. The core idea of the HTTP adaptive video streaming over mobile networks is that a video file is divided into consecutive segments and each segment is encoded to multiple discrete bitrates, then the suitable bitrate of video is se-lected based on the achievable data rate in the time varying wireless environment [5].

Since the technique of HTTP adaptive video streaming over mobile networks is proposed, the adaptive video transmission strategies in mobile networks to optimize the users’ QoE have been studied [6]-[8]. In [6], the issue of joint user association and rate allocation for

HTTP adaptive streaming in heterogeneous cellular networks is studied, then the authors model the optimization problem as a mixed integer programming problem, and joint user association and rate allocation based on the distributed greedy matching algorithm is pro-posed. The authors in [7] consider a stochastic QoS-aware Robust Predictive-DASH (RP-DASH) scheme over future wireless networks, and then the imperfect rate prediction is taken into account to achieve long-term quality fair-ness among the DASH users while capping the probability of service degradation. The problem of maximizing the experienced vid-eo quality at all the users is studied in [8] by jointly optimizing the time-domain resource partitioning (TDRP) for the HCN, the rate al-located to each specific user, and the selected video quality transmitted to a use. Then a pri-mal-dual approximation algorithm is designed to solve the optimization problem.

On the other hand, with a new promising technique named multi-access edge computing

(MEC) proposed by European Telecommuni-cations Standards Institute (ETSI), by using the MEC enhanced 5G networks to realize the video transmission has also been attracted a lot of attention [9]-[11]. This is because that the MEC can utilize the edge IT and cloud com-puting capabilities not only to perform video content placement during off-peak hours but also to realize the video transcoding, thereby smoothing out the temporal traffic variability,

reducing congestion and response latency, and improving the users’ QoE.

Taking these advantages into account, many works have been done for the HTTP adaptive video streaming transmission in MEC-based 5G networks recently [12]-[17]. The authors in [12] study the problem QoE-traffic opti-mization through collaborative edge caching in adaptive mobile video streaming, then a self-tuned bitrate selection algorithm with low complexity is designed to solve the opti-mization problem. A real-time, context-aware collaboration framework based on the MEC is studied in [13]-[14], then the authors pro-pose a cooperative caching and processing strategy called CoPro-CoCache for multi bit video streaming to enhance network resource utilization. An MEC enhanced adaptive bitrate (ABR) video delivery scheme is designed in [15] by combining content caching and ABR streaming technology together to make the cooperation between cache and radio re-source allocation. The authors of [16] study a QoE-driven mobile edge caching placement optimization problem for dynamic adaptive video streaming, then the different rate dis-tortion (R-D) characteristics of videos and the coordination among distributed edge servers are studied. In [17], the authors design an adaptive wireless video transcoding frame-work based on edge computing by deploying edge transcoding servers close to BSs, then an adaptive transcoding strategy is formulat-ed as a network utility maximization (NUM) problem and a low-complexity algorithm is proposed.

Although many works have been done for the dynamic adaptive video streaming in the MEC based 5G networks, most of them focus on the problem of caching or bitrate adaptation algorithm to optimize the network resource utilization or promote the users’ QoE, and the issue of the energy efficiency is largely ignored. However, the problem of energy-effi-cient video transmission is a key performance indicator in future 5G networks [18] due to the rapidly rising energy costs and increasingly rigid environmental standards. This is because

230 China Communications • July 2019

Authorized licensed use limited to: University of New South Wales. Downloaded on November 20,2020 at 02:28:37 UTC from IEEE Xplore. Restrictions apply.

that mobile video traffic enjoys the highest rate of growth and will account for seven-ty-eight percent of the total mobile data traffic by 2021, which will lead to an alarming rising in energy costs [19]. Therefore, to realize the energy efficient video delivery in MEC-based 5G networks is very important to meet the challenges raised by the high demands of traf-fic and energy consumption. Therefore, to fill this gap, based on our previous works [20], we give more detail consideration about the issue of energy-efficient joint caching and transcod-ing schedule strategy for HTTP adaptive video streaming in MEC based 5G networks in this paper. The main contributions of this paper can be summarized as follows:

* By utilizing the caching and computing capabilities of MEC at the edge of radio ac-cess network (RAN), we focus on the HTTP adaptive video streaming transmission in

MEC based 5G networks and consider the joint caching and transcoding by taking the distribution of the requested video segments into account. The issue studied in this paper can greatly alleviate the burden on backhaul link, whittle down the responding delay and improve the users’ QoE.

* Then, taking the energy efficiency as a key performance indicator, we study the prob-lem of joint optimization of caching, transcod-ing and transporting to achieve the minimum energy consumption of the network. Then three different modes for the process of video requesting and responding are considered, that is: *direct hit*, *transcoding hit* and *miss*. Especially, we go deep into the *transcoding* *hit* mode and arrange a transcoding scheduleto determine whether to respond users by transcoding or not, thus gains a good perfor-mance improvement than the strategies with-out transcoding schedule.
* Due to finding the optimal solution is NP-hard for an integer programming problem, which brings huge computation complexity. Therefore, to make the optimization problem tractable and reduce the computation com-plexity, a heuristic scheme based on simulated annealing (SA) algorithm is proposed to iter-

atively reach the global optimization of our formulation.

Finally, extensive simulation results are illustrated to demonstrate the performance of the proposed scheme based on SA algorithm. And numerical simulation results are shown that our proposed scheme has a lower com-plexity and higher accuracy than other classi-cal algorithms, such as the genetic algorithm and greedy algorithm.

The rest of the paper is organized as fol-lows. We first give the system model and problem formulation in Section II. Then, the joint caching and transcoding scheme based on a simulated annealing algorithm is designed In Section III. In Section IV, numerical sim-ulation results are illustrated and discussed. Finally, we conclude our paper in Section V.

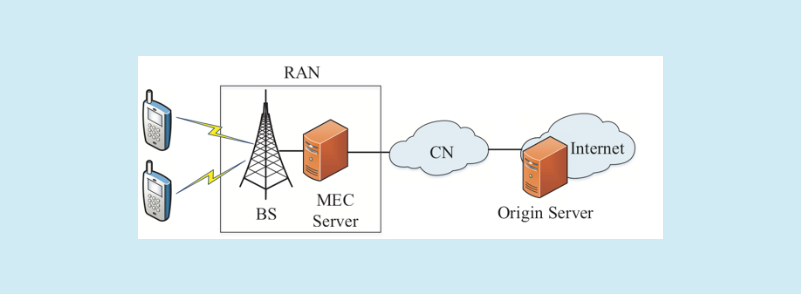
1. **SYSTEM MODEL AND PROBLEM**

**FORMULATION**

In this section, we first describe our system model and then the problem of joint caching and transcoding for HTTP adaptive streaming to realize the energy efficiency is formulated.

**2.1 System model**

As shown in Figure 1, we consider a 5G mo-bile network integrating the mobile edge com-puting (MEC) at the Radio Access Network (RAN). That is, the IT and cloud computing capabilities are introduced at RAN and the mobile computing, network control and stor-age are pushed to network edges. In this case, the services (i.e. video delivery etc.) can be quickly provided to the users, the quality of experience can be improved, and the network



**Fig. 1.** *The architecture of the 5G mobile networks with MEC.*

|  |  |
| --- | --- |
| China Communications • July 2019 | 231 |

Authorized licensed use limited to: University of New South Wales. Downloaded on November 20,2020 at 02:28:37 UTC from IEEE Xplore. Restrictions apply.

traffic through the backhaul link can be largely reduced.

Under this framework, we mainly take the HTTP adaptive video streaming delivery into account by using the computing and cach-ing capabilities of MEC in this paper. We assume that there are total *V* videos with a corresponding video set = {1,2,…, *v*,…,*V*} that can be requested by users. Each vid-eo *v* is divided into *Nv* consecutive video

segments with the corresponding set of segments as *v* = {*Sv*1 , *Sv* 2 ,…, *Svn* ,…, *SvNv* }, where the *Svn* denotes the *n* th segment of

video *v*. Then, we unify sets from 1 to *V* and define the set of all video segments as

= {*S*11 , *S*12 ,… , *S*1*N*1 ,… , *SV* 1 , *SV* 2 ,…, *SVNV* }. And to make the description simple, we redefine the set as {1,2,…, *s*,…, *S*} to denote the seg-

ments of all the videos, and *S* = ∑*V* 1 *N* is the

*v* = *v*

total number of video segments of all videos where the *s* denotes the *s* th segment in the whole segment set. In particular, the probabil-ity distribution vector of the segments can be denoted by = {*p*1 , *p*2 ,…, *ps* ,…, *pS* }, referred to as the segment popularity, where segment

*s* is requested with probability *p* and∑*S* 1 *p*

*s* *s* = *s*

=1.We also assume that each segment con-tains *L* seconds in playtime, and each segment can be encoded to *K* different bitrates and we define the set Ξ = {1,2,…, *k*,…, *K*} to denote

the possible bitrate versions. The bitrate ver-sions in set Ξ are arranged in ascending order, which means *k* = 1 represents the lowest bi-trate version and *k* = *K* represents the highest bitrate level. Let *Rsk* be the bitrate of the *s* th

segmentwith the *k* th bitrate version, and a higher bitrate than version *k* of the same seg-ment is defined as *Rsk* +. For simplicity, in the

remainderof this article, we also use *Rsk* and *Rsk* +to present the actual content of segment *s*

with bitrate version *k* and the exact content of segment *s* with a bitrate version higher than *k*, respectively. We assume that for each seg-ment, the bitrate is constant, thus, the size of

segment *s* with bitrate level *k* can be described as *L* \* *Rsk*.

Relying on the caching and computing ca-pabilities, the MEC server can not only cache different video segments with different bitrate versions but also can realize transcoding be-tween different bitrate versions of the same video segment. Hence, every time when video requests for *Rsk* is arrived at MEC server, there

would be three possible modes as inspired by [21]:

• *Direct hit* mode: when *Rsk* exactly exists

in the MEC server, then MEC server will re-spond to users directly by delivering *Rsk*.

• *Transcoding hit* mode: if *Rsk* dose not exist

in the MEC server, however there is a higher bitrate version of the segment *s*, denoted as *Rsk* +. In this case, the transcoding schedule may

be triggered. That is, MEC server can choose to transcode *Rsk* + to *Rsk*, or to request *Rsk*

from the origin content server through back-haul links, which depends on the scheme of transcoding schedule decision. Here, we notice that a higher bitrate version can be transcoded to a lower bitrate version, conversely, it is not able to upgrade a lower bit rate version to a higher bit rate one as proposed in [13].

* *Miss* mode: Neither *Rsk* exists, nor ahigher bitrate version *Rsk* + exists in the MEC

server. Under *miss* mode, the MEC server will respond to user requests by getting *Rsk* from

the origin server.

The above three possible cache response mode are described in Figure 2. As there are two optional ways to serve requests of us-ers under the *transcoding hit* mode, here we bring some more discussion on the necessity to do this choice, which we call “transcoding schedule”. Consider a scenario in which there is a request for *Rsk* = 500*kbps* and the MEC

server only holds a higher bitrate version *Rsk* +=5*Mbps*. If the MEC server chooses to

transcode *Rsk* + to *Rsk*, the difference of segment

size between the higher bit rate version and the requested bit rate version can be presented

232 China Communications • July 2019

Authorized licensed use limited to: University of New South Wales. Downloaded on November 20,2020 at 02:28:37 UTC from IEEE Xplore. Restrictions apply.

as (*Rsk* + − *Rsk* ) \* *L*, which is much greater than the data size of retrieving *Rsk* from origin serv-

er. And transcoding cost is determined by three factors: input bit-rate, target bit-rate, and video length, therefore, here we use (*Rsk* + − *Rsk* ) \* *L*

to present the data size involved to be dealt with by the MEC server. Thus, there is a tradeoff between transcoding cost and retriev-ing cost and it is wise to arrange a transcoding schedule to enhance the network performance.

Under this setup, taking the caching and transcoding mechanism into consideration, two main problems should be carefully solved to optimize the network performance: content placement and transcoding schedule, under the constraints of the caching capacity and com-puting capability of MEC server. Due to the energy efficiency is one of the most important key performance index in 5G networks, we mainly take the energy efficiency into account, and consider the joint caching and transcoding to improve the energy efficiency in this paper.

**2.2 Problem formulation**

Based on the above assumption, we will for-mulate the optimization problem for the ener-gy-efficient joint caching and transcoding in this subsection. Before given the optimization problem, we first give some assumptions for the variables and constants used in the prob-

lem formulation which are summarized in Table I, then the detailed explanation of vari-ables and constants are as follows. Let *xsk* be

the caching variable that indicates the decision of whether to cache the video segment *s* with version *k*. And *xsk* = 1 denotes that the video

segment *s* with version *k* is cached at MEC server, otherwise, *xsk* = 0.

We also assume that the MEC server has the maximum caching capacity *C* *m* (*GB*). In

this case, the total caching size of MEC server could not exceed the maximum caching capac-ity, which can be expressed as follows.

|  |  |
| --- | --- |
| *S K* |  |
| ∑∑ *xsk* \* *L* \* *Rsk* ≤ *Cm*. | (1) |

*s* =1 *k* =1

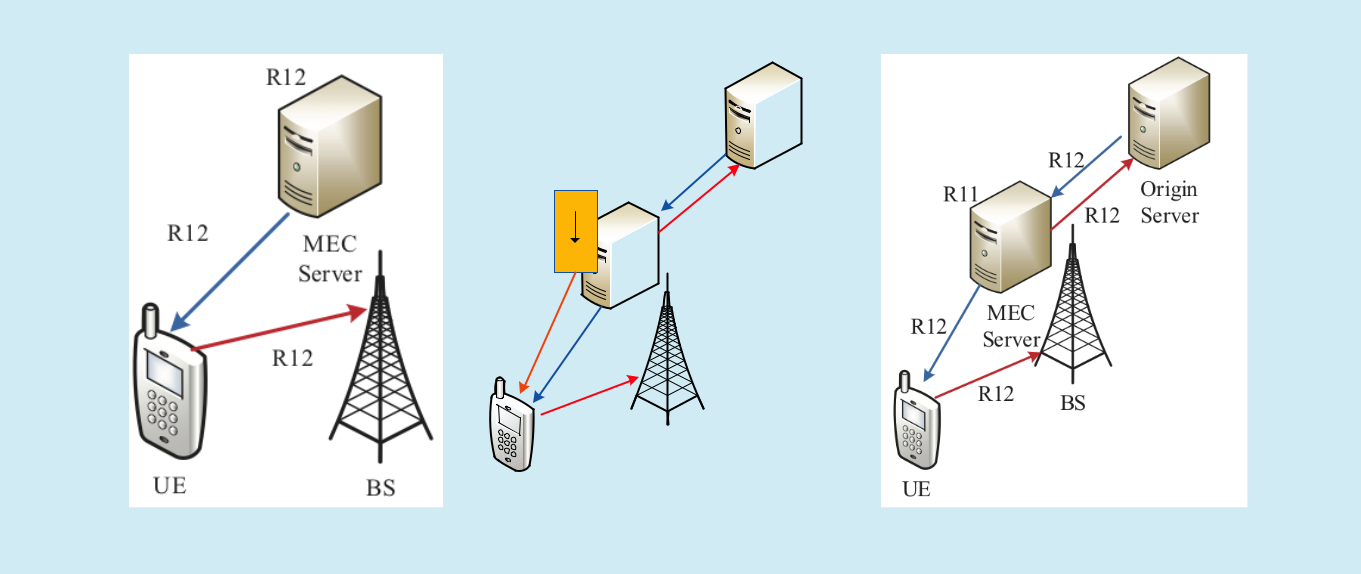
As the MEC server can provide the com-puting capability to realize transcoding be-tween different bitrate versions of the same video segment, we can let *zsk* + be the variable

to indicate whether there exists a higher bitrate version than version *k* of segment *s* cached in MEC server. We set *zsk* + = 1 when MEC server

caches a higher version of segment *k*, *zsk* + = 0

otherwise. Therefore, we can easily draw a conclusion that the transcoding procedure possibly be executed only when there is a seg-ment with higher bitrate version cached in the MEC server, so *zsk* + and *xsk* should follow the

following constraint.



|  |  |  |
| --- | --- | --- |
| **7UDQVFRGLQJ** | **5** |  |
| **2ULJLQ** |  |
| **5** |  |
| **5 6HUYHU** |  |
| **5** |  |
| **5** |  |  |

**5** **0(&**

**5 6HUYHU**

**5**

**%6**

**8(**

(a) Direct hit mode (b) Transcoding hit mode (c) Miss mode

**Fig. 2.** *Three possible cache hit modes.*

|  |  |
| --- | --- |
| China Communications • July 2019 | 233 |

Authorized licensed use limited to: University of New South Wales. Downloaded on November 20,2020 at 02:28:37 UTC from IEEE Xplore. Restrictions apply.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Table I.** *Main parameters used in the formulation.* | |  |  |  |  | the energy consumption of the 5G networks. | |  |
| Symbol | Description |  |  |  |  | And there are the following three kinds of en- | |  |
|  |  |  |  |  |  | ergy consumption should be considered in our | |  |
| *S* | The total number of video segments |  |  |  |  |  |
| *L* | The playtime length of a video segment, (s) |  |  |  |  | problem formulation. |  |  |
| *K* | The total number of available video bitrate versions | |  |  |  | • Caching Energy *Ec*: This component is | |  |
| *Rsk* | The video segment *s* with bitrate version *k*, as well as the exact bitrate | | | | | the energy required to store the contents to the | |  |
| (Mbps) of segment *s* with bitrate version *k* |  |  |  |  | MEC server, which is directly proportional to | |  |
|  |  |  |  |  |  |
| *Rsk* + | A video segment *s* with a higher bitrate version than version *k*, as well as | | | | | the data size stored in MEC server. Here we | |  |
| the exact bitrate (Mbps) of segment *s* with a higher bitrate version than *k* | | | | | adopt an energy proportional model [22]-[23]. | |  |
|  |  |
| *Cm* | The caching capacity of MEC server, (GB) |  |  |  |  |  |
|  |  |  |  | Assume that the caching energy required for | |  |
| *wcm* | The caching energy required for storing each bit of the content in MEC | | | | | storing each bit of content in an MEC server | |  |
| server per unit time, (J/bit) |  |  |  |  |  |
|  |  |  |  |  | during per unit time is *wcm* (*J* / *bit* ), then the | |  |
|  | The number of CPU cycles required to process one bit input of MEC | | | | |  |
| *cm* | energy consumed by caching *Rsk* is given by: | |  |
| server per unit time, (cycles/bit) |  |  |  |  |  |
|  |  |  |  |  |  |
| *δ* | The energy efficiency for MEC server to execute one CPU cycle, (Wat- | | | | | *Ec* = *wcm* \* *L* \* *Rsk*. | (4) |  |
| t/GHz) |  |  |  |  |  |  |  |
|  |  |  |  |  | • Transport Energy *Et*: Notice the fact that | |  |
| *f*0 | The dominant frequency of CPU, (GHz) |  |  |  |  |  |
|  |  |  |  | it is an inevitable process to transport content | |  |
| *e*0*m* | The transmission energy efficiency of the backhaul link between MEC | | | | |  |
| from MEC server to users through wireless | |  |
| sever and origin server, (Watt/bit) |  |  |  |  |  |
|  |  |  |  |  | links under the fore-mentioned three cache | |  |
| *Tt* | The delay caused by retrieving *Rsk* for the origin server, (s) | | |  |  |  |
|  |  | hit mode, which means that this part contrib- | |  |
| *Td* | The delay caused by trancoding *Rsk* +to *Rsk*, (s) |  |  |  |  |  |
|  |  |  |  | utes the same amount of energy consumption | |  |
| *Ec* | Caching energy consumption, (J) |  |  |  |  |  |
|  |  |  |  | to the total energy consumption of the three | |  |
| *Et* | Transporting energy consumption, (J) |  |  |  |  |  |
|  |  |  |  | requesting-responding modes. For simplicity, | |  |
| *Ed* | Transcoding energy consumption, (J) |  |  |  |  |  |
|  |  |  |  | we ignore this wireless transporting part and | |  |
| *xsk* | Takes the value of 1 if *Rsk* is cached in MEC server and 0 otherwise | | |  |  |  |
|  |  | focus on the transporting energy consumption | |  |
|  | Takes the value of 1 if MEC server chooses to transcode *Rsk* +to *Rsk* and 0 | | | | |  |
| *ysk* | caused by retrieving content from the origin | |  |
| if *Rsk* is retrieved from origin server |  |  |  |  |  |
|  |  |  |  |  | server to the MEC server through backhaul | |  |
| *zsk* + | Takes the value of 1 if *Rsk* +is cached in MEC server and 0 otherwise | | |  |  |  |
|  |  | links. Assuming that the transporting energy | |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  | efficiency and the bandwidth of the backhaul | |  |
|  |  | *K* |  |  |  | link between the MEC server and the origin | |  |
|  | (2) | | server are *eom* (*Watt* / *bit*) and *Wom* (*Mbps*)[22], | |  |
|  |  |  |
|  | *z sk* +=min 1, | ∑ *xsi* . | |  |
|  |  | *i* = *k* +1 |  |  |  | respectively, then we can get the transporting | |  |
|  | In the above assumption,*xsk* and *zsk* + are | | | | | delay through the following formula [23]: |  |  |
|  | the binary variables to indicate the content | | | | | *Tt* = *L* \* *Rsk* / *Wom*, | (5) |  |
|  | placement decision. As for the transcoding | | | | | then, considering the contribution as pro- | |  |
|  | schedule decision, here we use binary variable | | | | |  |
|  | portional with respect to the amount of data | |  |
|  | *ysk* to describe. And *ysk* | takes 1 if MEC server | | | |  |
|  | transported by the network [24], we give the | |  |
|  | chooses to transcode *Rsk* + to *Rsk* and 0 if *Rsk* is | | | | |  |
|  | transporting energy consumption as follows: | |  |
|  | retrieved from the origin server. Consider that | | | | |  |
|  | *Et* = *eom* \* *L* \* *Rsk* \**Tt*. | (6) |  |
|  | the transcoding schedule is valid only when a | | | | |  |
|  | • Transcoding energy *Ed*: Assuming that | |  |
|  | content request corresponds to the *transcode* | | | | |  |
|  | *hit* mode, *ysk* subjects to the following con- | | | | | the number of CPU cycles required to process | |  |
|  | straint: |  |  |  |  | one bit input is *cm* (*cycles* / *bit*), thus, the time | |  |
|  | *ysk* ≤ *zsk* +. | |  | (3) | | consumed by transcoding can be given by [24] | |  |
|  | Then from an energy efficiency perspective, | | | | | [25]: |  |  |
|  | *Td* = *cm* \* *L* \*(*Rsk* +− *Rsk* )/ *f*0. | (7) |  |
|  | our objective of proposing the joint caching | | | | |  |
|  |  |  |  |
|  | and transcoding schedule policy is to minimize | | | | | where *f* 0 (*GHZ* ) and the numerator in the above | |  |
|  |  |  |  |  |  |  | |  |
|  |  |  |  |  |  |  | |  |
| 234 |  |  |  |  |  | China Communications • July 2019 | |  |

Authorized licensed use limited to: University of New South Wales. Downloaded on November 20,2020 at 02:28:37 UTC from IEEE Xplore. Restrictions apply.

formula are the dominant frequency of CPU and the total CPU cycles required to transcode *Rsk* +to *Rsk*, respectively. Then, inspired by [26],

the transcoding energy consumption can be given:

|  |  |
| --- | --- |
| *Ed* = *δ* \**cm* \* *L* \*(*Rsk* +− *Rsk* )\**Td*, | (8) |

where *δ* (*Watt* / *GHz*) is the energy efficiency for MEC server to execute one CPU cycle.

According to the above analysis and taking the request probability distribution into con-sideration, the average total energy consump-tion of serving the video requests can be given by the following formula:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *S* | *K* |  |  |  |  |  |  |  |  |
| *Etot* =∑ *p s* ∑ *xsk Ec* + | | | |  |  |  |  | . |  |
| *s* =1 | *k* =1 | |  |  |  |  |  |  |
| (1− *x* ) (1− *z* | | *sk* | + )*E* + *z* | *sk* | + [ *y* | *E* +(1− *y* | *sk* | )*E* ] |  |
| *sk* { |  | *t* |  | *sk d* | *t* } |  |

transcoding should not exceed the finite capacity of the CPU. The third constraint

*K*

*z sk* += *min*{1,∑ *xsk* }indicates the availability

*i* = *k* +1

of a higher bitrateversion for transcoding, in which *zsk* + is highly coupled with *xsk*.The fourth

constraint *y* *sk* ≤ *zsk* + provides a precondition for

the availability of transcoding schedule. The last three formulas expose binary constraint to the indications of the content placement and transcoding schedule respectively.

1. **THE SOLUTION TO THE JOINT CACHING AND TRANSCODING FOR ENERGY EFFICIENCY**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  | (9) | In this section, we will solve the joint caching |  |
|  | To achieve an energy-efficient network | | | | | and transcoding problem for energy efficien- |  |
| system, we now formulate the optimization | | | | | | cy formulated in (10). From the optimization |  |
| problem as the following integer linear pro- | | | | | | problem in (10), we can observe that the |  |
| gramming (ILP) problem: | | | | |  | problem of joint caching and transcoding is |  |
|  |  |  | min | *Etot*. | (10) | an integer linear programming problem (ILP) |  |
|  |  |  | *xsk* , *y sk* ,*zsk*+ | |  | and finding the optimal solution is an NP-hard |  |
| Subject to:` | | |  |  |  |  |
|  |  |  | problem, which may bring huge computation |  |
|  | *S* | *K* |  |  |  |  |
|  |  | ≤ *C m* ,∀*s* ∈, *k* ∈Ξ, |  |  |  |
|  | ∑ ∑ *xsk* \* *L* \* *Rsk* | | | (11) | complexity. Therefore, in order to reduce |  |
| *S* | *s* =1 | *k* = |  |  |  | the computation complexity, an approximate |  |
| *K* |  |  |  |  |  |  |
| ∑ ∑ *y* *sk* \* *L* \*(*Rsk* + − *Rsk* )\**cm* ≤ *f* 0 ,∀*s* ∈ , *k* ∈ Ξ | | | | | | method based on simulated annealing (SA) |  |
| *s* =1 | *k* = |  |  |  | (12) | algorithm is proposed to obtain the best solu- |  |
| , |  |  |  |  | tions for our problem in this section. We first |  |
|  |  |  | *K* |  |  |  |
|  | *z sk* + |  |  |  |  |  |
|  | = min{1, ∑ *xsi* },∀*s* ∈ , *k* ∈ Ξ, | | | (13) | give a brief introduction for the basic princi- |  |
|  |  |  | *i* = *k* +1 |  |  | ple of the simulated annealing in Subsection |  |
|  |  | *y sk* ≤ *z sk* +,∀*s* ∈, *k* ∈ Ξ, | | | (14) | III-A. Then the joint caching and transcoding |  |
|  |  | *xsk* | ∈ {0,1},∀*s* ∈ , *k* ∈ Ξ, | | (15) | scheme for energy efficiency based on the |  |
|  |  | *y sk* | ∈ {0,1},∀*s* ∈ , *k* ∈ Ξ, | | (16) | simulated annealing algorithm is proposed in |  |
|  |  | *z sk* +∈{0,1},∀*s* ∈, *k* ∈ Ξ. | | | (17) | Subsection III-B. |  |

The objective function expects to minimize the total energy consumption of the system under the constraints. And the first constraint

*S K*

∑∑*xsk* \* *L* \* *Rsk* ≤ *Cm* that the total content

*s* =1 *k* =1

size cached in the MEC server should below the constraint of caching capacity, the sec-

*S K*

ond constraint ∑∑ *y* *sk* *L* (*Rsk* + − *Rsk* )*cm* ≤ *f*0

*s* =1 *k* =1

denotes that the computation using the

**3.1 The basic principle of the**

**simulated annealing**

Simulated annealing is a classical optimization technique to solve combinatorial optimization problems, which is derived from annealing in metallurgy, a technique involving heating and controlled cooling of a material to increase the size of its crystals and reduce its defects [27].

Using the physical annealing process as an analogy, the working process of simulated an-nealing applied to combinatorial optimization

|  |  |
| --- | --- |
| China Communications • July 2019 | 235 |

Authorized licensed use limited to: University of New South Wales. Downloaded on November 20,2020 at 02:28:37 UTC from IEEE Xplore. Restrictions apply.

problems can be described as follows: It starts by setting a random initial feasible solution as the initial state and computes its correspond-ing profit by applying the objective function, which is energy in physical annealing. The control parameter *T* is also set at a relative high value, which will gradually decrease to zero in the search process under the control of a given annealing *factor* that is advised to take a value between 0.8 and 0.99 [28]. At each state, a neighboring solution will be calcu-lated by using a well defined function named *generate*. Then if the profit difference of thetwo states leads to profit decrement, the new neighboring solution is accepted. Otherwise, it is accepted by a probability which is defined as Boltzman’s distribution PDF, which can ef-

**Table II.** *Relationship between physical annealing and simulated annealing used**in combinatorial optimization.*

|  |  |  |
| --- | --- | --- |
| Physical annealing | General simulated annealing | Our proposed scheme based on SA |
|  |  |  |
| System States | Feasible Solutions | ***x, y*** and ***z*** |
| Energy | Objective Function profit | *Etot* in the formula (9) |
| Change of State | Neighboring Solutions | ***x***\****, y***\* and ***z***\* |
| Temperature | Control Parameter | Temperature *T* |
| Frozen State | Optimum Solution | Optimum Solution |
|  |  |  |

**Algorithm 1.** The joint caching and transcoding for energy efficiency based on thesimulated annealing algorithm.

1. **begin**
2. *T* ← *max\_temper* ,***x***←0,***y***←0,***z***←0
3. **while**(*T*>*min\_temper*) **do**
4. get current energy consumption *Etot* , using ***x***, ***y***, ***z***
5. generate new set of solution ***x\****, ***y\****, ***z\****
6. get new energy consumption *Etot*\*,, using ***x\****, ***y\****, ***z\****
7. evaluate the difference between the new and the current energy consumption by ∆*E* = *Etot*\* − *Etot*
8. **if**(∆*E*<0) then
9. accept ***x\****, ***y\****, ***z\**** as the current solution
10. **else**
11. **if**(*e*−∆*E*/*T*>*r*and (1)) **then**
12. accept ***x\****, ***y\****, ***z\**** as the current solution
13. **end if**
14. **end if**
15. *T* ← *T \* factor*
16. **end**
17. **End**

fectively avoid falling into the local minimum

1. and will be justified in the Section IV. At last, the algorithm will achieve the optimum solution of the combinatorial optimization problem, which is called the frozen state in physical annealing.

**3.2 The joint caching and transcoding for energy efficiency based on the simulated annealing algorithm**

According to the above description and based on the optimization problem in (10), we can first design the relationship among our pro-posed scheme based on simulated annealing and physical annealing, as well as the general simulated annealing used in common combi-natorial optimization problems, which is sum-marized in Table II. In our proposed scheme based on SA algorithm, the binary variables ***x, y*** and ***z*** denote a set of feasible solutions ingeneral simulated annealing and a system state in physical annealing, while the set of ***x***\****, y***\* and ***z***\* denotes a set of neighboring solutions and indicates a change of state in physical annealing. Meanwhile, *Etot* in the formula (9)

corresponds to the objective function profit in general simulated annealing, which is an anal-ogy concept of energy in physical annealing. In our proposed scheme based on SA algo-rithm, we still adopt the temperature *T* as the control parameter and the optimum solution in our scheme and the traditional SA corresponds to the frozen state in physical annealing.

Then combining our proposed formulation with the above physical annealing process and simulated annealing process, the structure of our proposed scheme based on SA algorithm is described as Algorithm 1.

Algorithm 1 starts by setting a set of initial solution on ***x, y*** and ***z***, where ***x, y*** and ***z*** denote a possible combination of *xsk*, *ysk* and *zsk* + ,

respectively. The control parameter, tempera-ture *T*, is also initialised to a reasonable high value, *max\_temper*, which will eventually decrease to *min\_temper* in the search process under the control of a given annealing *factor*.

236 China Communications • July 2019

Authorized licensed use limited to: University of New South Wales. Downloaded on November 20,2020 at 02:28:37 UTC from IEEE Xplore. Restrictions apply.

Every time there generates a new set of neigh-boring solutions, ***x***\****, y***\* and ***z***\*, the algorithm computes the energy consumption difference, denoted as ∆*E* < 0, between the new solution and the current solution. If the difference val-ue is negative, then accept the new solution as the current solution, otherwise, accept the new solution if *e* −∆*E* /*T* > *r* and (1), which avoids getting stuck in a local minimum.

As the Algorithm 1 describes the basic framework of our iteration process, we can see that an important premise to push the iteration process is generating a new neighboring solu-tion subjecting to the constraints from formula

1. to (17) at each state, which also prevents the occurring of useless solutions that do not obey our formulation assumptions. Therefore, how to generate the new neighboring solution is a key step in the Algorithm 1. Based on this, we design the procedure of SA generate to randomly generate the new neighboring solu-tion, ***x***\****, y***\* and ***z***\*, from the feasible data set near the current feasiblesolution ***x, y*** and ***z*** un-der the constraints from formula (11) to (17).

The structure of the procedure of SA generate is described in Algorithm 2.

In a fixed state, the time complexity to generate a new neighboring solution of our proposed scheme based on SA algorithm can be described as *O* (*S* \* *K* ), and the total num-

ber of states has close relationship with the control parameter *T* and the annealing *factor*. The higher the initial value of *T* and the an-nealing *factor*, the higher the global search capacity of our proposed scheme and the more time it needs to find the optimum solution, conversely, the time complexity will decrease but the global search capacity will be degrad-ed if the initial value of *T* and the annealing *factor* decrease. That is, taking appropriateparameter combination, the algorithm will be characterized by a relatively low time com-plexity while achieving an ideal convergence. The innovative feature of our scheme based on SA to accept the ”bad solution” by a probabil-ity can effectively avoid falling into the local minimum and eventually achieve the global

optimum approximate solution [29], which

**Algorithm 2.** Procedure of SA generate.

**Input: *x***, ***y*** and ***z***

**Output: *x\****, ***y\**** and ***z\****

1. **begin**
2. *overfilled* ← *false* , *overComputed* ← *false* , *flag* ← *false*
   1. *K*
3. **if**(∑∑*x**sk*\**L*\**Rsk*>*Cm*) **then**

*s* =1 *k* =1

1. *overfilled* ← *true*
2. **end if**
3. **While**(not*overfilled)* **do**
4. Select a *xsk* randomly from *x* which equals 0, and let *xsk* ← 1
   1. *K*
5. **if**(∑∑*x**sk*\**L*\**Rsk*>*Cm*) **then**

*s* =1 *k* =1

1. *overfilled* ← *true*
2. **end if**
3. **end**
4. **While**(*overfilled)* **do**
5. Select a *xsk* randomly from *x* which equals 1, and let *xsk* ← 0
   1. *K*
6. **if**(∑∑*x**sk*\**L*\**Rsk*≤*Cm*) **then**

*s* =1 *k* =1

1. *overfilled* ← *false*
2. **end if**
3. **end**
4. *y* ←0, *z* ←0
5. **for**(s in 1:S) **do**
6. **for**(*k*in 1:*K*) **do**

|  |  |
| --- | --- |
| 21. | *zsk* +←min{1,∑*iK*= *k* +1 *xsi*} |
| 22. | **if** (*y**sk*>*zsk*+) **then** |
| 23. | *flag* ← *true* |
| 24. | **end if** |
| 25. | **if** (*flag*) **then** |
| 26. | *ysk* ← *0* |
| 27. | **end if** |
| 28. | **end** |
| 29. | **end** |
|  | *S K* |
| 30. | **if (**∑∑*y**sk*\**L*\*(*Rsk*+−*Rsk*)\**cm*>*f*0) **then** |
|  | *s* =1 *k* =1 |

1. *overComputed* ← *true*
2. **end if**
3. **While(***overComputed***)do**
4. select a *ysk* randomly from **y** which equals 1, and let *ysk* ← *0*

|  |  |
| --- | --- |
|  | *S K* |
| 35. | **if (**∑∑*ysk*\**L*\*(*Rsk*+−*Rsk*)\**cm*≤*f*0) **then** |
|  | *s* =1 *k* =1 |

1. *overComputed* ← *false*
2. **end if**
3. **end**
4. **End**

|  |  |
| --- | --- |
| China Communications • July 2019 | 237 |

Authorized licensed use limited to: University of New South Wales. Downloaded on November 20,2020 at 02:28:37 UTC from IEEE Xplore. Restrictions apply.

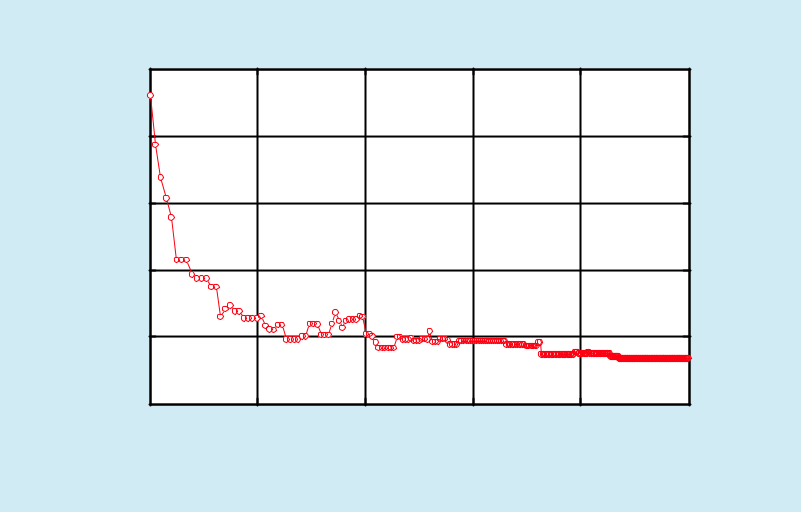
outperforms the traditional algorithm such as greedy algorithm. On the other hand, com-pared to the genetic algorithm which depends on population integrity search and needs to encode and decode for the population, the pro-posed scheme based on simulated annealing algorithm searches the optimum solution by a single individual and omits the encoding and decoding process, thus is very simple and can be used for the optimization of complex non-linear problems.

**IV. SIMULATION RESULTS AND**

**DISCUSSIONS**

In this section, we use the computation simu-lation method to demonstrate the performance of the proposed scheme. On the one hand, we reveal the key system parameters that affect the performance of our proposed video de-livery strategy, on the other hand, we provide the performance comparison between our proposed joint caching and transcoding strat-egy and another two strategies. Furthermore, we provide illustrations of the performance advantages of our proposed scheme based on simulated annealing algorithm by comparing to the traditional greedy algorithm and random algorithm as proposed in [24].

In our simulation, the simulation pa-rameters are set as follows. We investi-



|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | 5000 |  |  |  |  |  |  |
| (J) | 4000 |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
| Consumption | 3000 |  |  |  |  |  |  |
| 2000 |  |  |  |  |  |  |
| Energy |  |  |  |  |  |  |
| 1000 |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  | 0 | 80 | 60 | 40 | 20 | 0 |  |
|  | 100 |  |
|  |  |  | Current Temperature (oC) | |  |  |  |

**Fig. 3.** *Temperature impact on the convergence rate of the proposed scheme based**on simulated annealing algorithm.*

gate an MEC server with *f* 0 = 3.4*GHz* , *cm* =300*cycles* / *bit*[30] and *δ* =1*Watt* / *GHz*

1. The caching capacity of MEC server is

*C m* =5*GB* and the caching energy efficien-cy is *wcm* = 8\*10−8 *J* / *bit* [22]. Bandwidth

and the transporting energy efficiency of the backhaul link are given by *Wom* = 30*Mbps* and

*eom* =8\*10−6*Watt* / *bit*[22], respectively. As for

the video segment, we set *S* = 300, *L* = 10*s* and *K* =4, unless otherwise specified. We also as-sume that the popularity of the segments is ar-ranged in a descending order according to the Zipf’s law, the request probability of segment

*s* can be writtenas *p* = (1/ *sα* ) ,∀*s* ∈ ,

*s* ∑*iS*=11/ *iα*

where *α* ∈[0.5,1.5] is the skewness re -

flecting the concentration of the popularity distribution. In addition, we provide five available sets of bitrate versions with dif-ferent intervals, listed as [1,1,1,1]Mbps, [1,6,11,16]Mbps , [1,11,21,31]Mbps , [1,16,31,46]Mbps, [1,21,41,61]Mbps.

Experimental results show that the setting of the control parameter *T* and the annealing *factor* influence the global search performanceof our proposed scheme significantly [29], thus the parameters must be carefully taken and the convergence of our proposed scheme must be verified before we actually begin our simulation, which is the basis of our whole simulation. Figure 3 shows the relationship that energy consumption with the decreasing of control parameter, temperature. To reach the convergence, we set the start temperature at 100, and the annealing *factor* at 0.99, while the end temperature is set at 0.05 [28]. In Figure 3, we can observe that the tendency of convergence rate is shown clearly by the results of the energy consumption. In the beginning, energy consumption decreases fast, and with the temperature decreasing at a relatively low-er level, it displays a premature convergence and then maintains at a nearly constant value. Our next simulations are carried out at the ba-sis of the setting of above two parameters. It

238 China Communications • July 2019

Authorized licensed use limited to: University of New South Wales. Downloaded on November 20,2020 at 02:28:37 UTC from IEEE Xplore. Restrictions apply.

is worth noting that Figure 3 also validates the capability of our proposed scheme based on SA algorithm to jump out of a local minimum and approximate the global optimal result.

By verifying the convergence of the pro-posed scheme in Figure 3, now in our next simulation, we will first focus on the three strategies which are described as follows:

* Our proposed scheme, which takes the transcoding schedule into consideration under the *transcode hit* mode. In the other words, when there occurs a *transcode hit*, MEC serv-er can respond to users by bitrate transcoding or requesting corresponding segment from the origin server, which depends on the result of energy consumption optimization.
* Caching and transcoding, which means there is no trans-porting under *transcode hit* mode. Every time when a *transcode hit* oc-curs, MEC server transcodes a higher bitrate version to the requested one, no matter how much the data size it need to process [14].
* Caching and transporting, which means there is no transcoding under *transcode hit* mode. Every time when a *transcode hit* occurs,

MEC server responds the request by retrieving corresponding segment from the origin server, which indicates that the MEC server does not hold the transcoding capability anymore. This strategy actually corresponds to the video de-livery in traditional cellular network without

MEC [31].

In Figure 4, we investigate the impact of the playtime length of video segments on the energy consumption and [1,10,30,60]*Mbps* is

given as the set of segment bitrate ver-sions. As formulas of (4), (7) and (8) show, *Ec*, *Et*,*Ed*

are linear in *L*, as a result, their product,*Etot*,

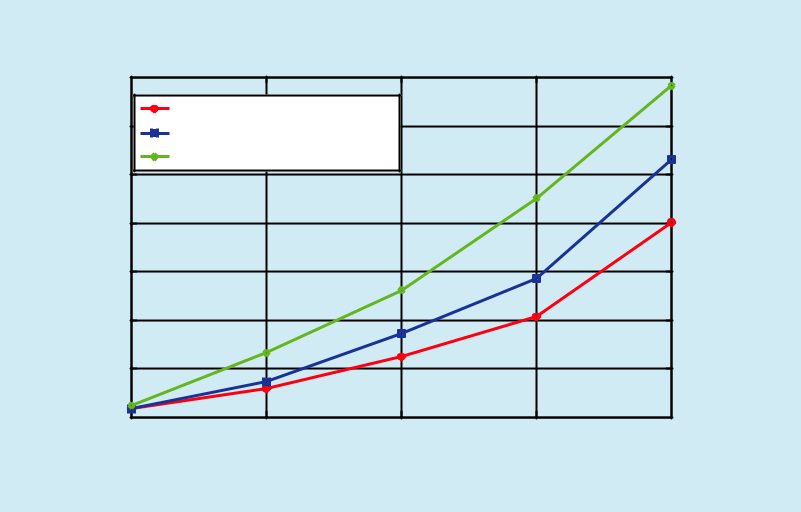
is quadratic in *L*, which is responsible for the shape of the curves in Figure 4. The red curve presents the energy consumption under our proposed formulation, while the blue curve describes the energy consumption without transporting under transcode hit mode and the green curve depicts the energy consumption without transcoding under *transcode hit* mode. All of the three curves show an upward trend

with the increase of segment playtime length, due to the increasing of data size of each video segment essentially. When the playtime length is relatively short, the data size of the video segment, *L* \* *Rsk*, is small. Correspondingly, the

difference between different bitrate versions of the same video chunk *L* \* (*Rsk* + − *Rsk* ), is relatively small as well. As a result, transcod-ing process can be done by high-performance MEC server in a very short time period with lower energy consumption than retrieving con-tent segment through the time-consuming and energy-consuming backhaul link. Thus, the red curve and the blue curve tend to overlap with each other when the playtime length is short, as both of the two strategies would choose to respond to users by transcoding when there occurs a *transcode hit*. In addition, the third strategy that without transcoding under *transcode hit* mode yields more energy con-sumption than the other two strategies, which is not obviously presented due to the limit of data size when the playtime length is rela-tively short. With the increasing of playtime length *L* \*(*Rsk* + − *Rsk* ) would be much greater

than *L* \* *Rsk*, which would cause higher ener-

gy consumption if we insist on transcoding a higher bitrate version to a lower one. Thus, it is wise to introduce a transcoding schedule to minimize the energy consumption. Our sim-ulation results also confirm this analysis, and



|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | 7 x 104 |  |  |  |  |  |
|  | 6 | our proposed scheme |  |  |  |  |
|  | caching and transcoding |  |  |  |  |
| (J) |  |  |  |  |  |
| 5 | caching and transporting |  |  |  |  |
| Consumption |  |  |  |  |  |
| 4 |  |  |  |  |  |
| 3 |  |  |  |  |  |
| Energy | 2 |  |  |  |  |  |
|  |  |  |  |  |  |
|  | 1 |  |  |  |  |  |
|  | 02 | 4 | 6 | 8 | 10 |  |
|  |  | Playtime Length of Each Video Segment (s) | | |  |  |

**Fig. 4.** *Playtime length of video segments impact on energy consumption.*

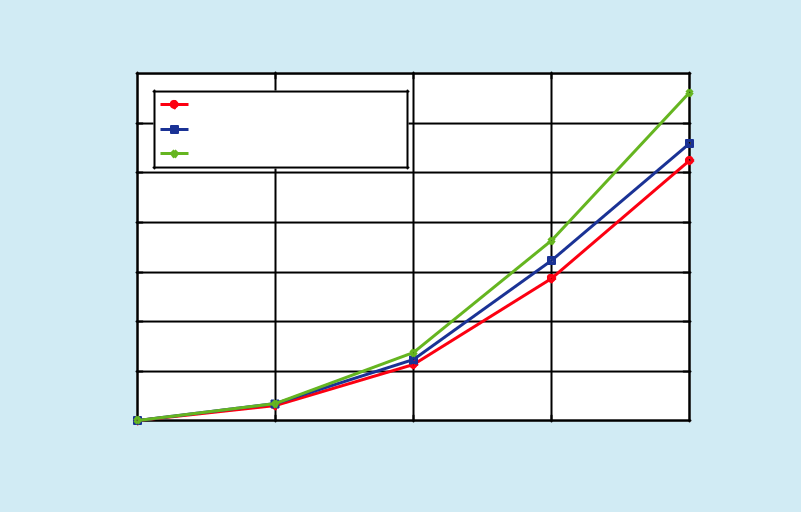
|  |  |
| --- | --- |
| China Communications • July 2019 | 239 |

Authorized licensed use limited to: University of New South Wales. Downloaded on November 20,2020 at 02:28:37 UTC from IEEE Xplore. Restrictions apply.

the experimental data also indicates that our transcoding schedule strategy outperforms the other two strategies and brings an approximate six percent performance improvement.

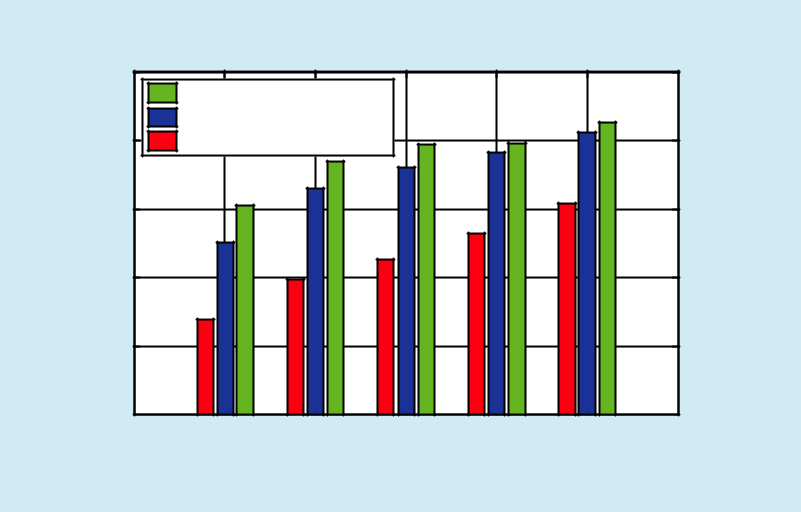
As the interval between bitrate versions re-fers to the bitrate difference of the same video clip between a higher bitrate and a lower bi-trate, which can be described as *Rsk* + − *Rsk*, this

value directly affects the data size needed to be proceed by MEC server when a *transcode* *hit* occurs. In Figure 5, we use five sets of bi-trate versions with different intervals 0*Mbps*, 5*Mbps*, 10*Mbps*, 15*Mbps*, and 20*Mbps*, list-e d a s [1,1,1,1]*Mbps* , [1,6,11,16]*Mbps* ,



|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | 7 x 104 |  |  |  |  |  |
|  | 6 | our proposed scheme |  |  |  |  |
| (J) | caching and transcoding |  |  |  |  |
| 5 | caching and transporting | |  |  |  |
| Consumption |  |  |  |  |  |
| 4 |  |  |  |  |  |
| 3 |  |  |  |  |  |
| Energy | 2 |  |  |  |  |  |
|  |  |  |  |  |  |
|  | 1 |  |  |  |  |  |
|  | 00 | 5 | 10 | 15 | 20 |  |
|  |  | Interval between Video Bitrate Versions (Mbps) | | |  |  |

**Fig. 5.** *Interval between bitrate versions impact on energy consumption.*



|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 2.5 x 104 | caching and transporting | |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  | 2 | caching and transcoding | |  |  |  |  |  |
| Consumption(J) | our proposed scheme | |  |  |  |  |  |
| 1.5 |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
| Energy | 1 |  |  |  |  |  |  |  |
| 0.5 |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  | 00 | 100 | 200 | 300 | 400 | 500 | 600 |  |
|  |  |  | Number of Video Segment | | |  |  |  |

**Fig. 6.** *Number of video segment impact on energy consumption.*

240

[1,11, 21,31]*Mbps* , [1,16,31,46]*Mbps* a n d

[1,21,41,61]*Mbps* to investigate the relation-ship between energy consumption and the bi-trate interval. Similar to the analysis of Figure 4, when the interval is small, its’ influence on data size is negligible, so the red curve and the blue curve tend to overlap with each other. As the interval increasing, it is unwise to always carry out a bitrate transcoding or a transport-ing form origin server without considering a transcoding schedule. Simulation results in Figure 5 show that our transcoding schedule strategy has obvious performance improve-ment when the bitrate version interval grows large than the other two strategies. Of course, the energy consumption of the three schemes increase as the bitrate interval increases.

In Figure 6, we study the impact of the number of video segment on the energy con-sumption under the three strategies. As Figure 6 shows, energy consumption increases as the number of video clips increases. When the number of video segments is small, the system tends to cache all the video segments due to the affluent storage space and the cheap caching energy efficiency. Therefore, *direct* *hit* happens with a very large probabilitywhile *miss* and *transcode hit* may be absent. As a result, energy consumptions under the three strategies are approximately the same. As the number of video segments increases, the capacity-constrained MEC server would cache only a part of video segments, so *miss* and *transcode hit* occur inevitably with a large probability. As analysis of Figure 4 and Fig-ure 5, our proposed transcoding schedule will outperform the other two strategies without transcoding schedule, which has been proved by our experimental results in Figure 6.

Figure 7 shows the relationship between caching capacity of MEC server and the en-ergy consumption under the three different strategies. When caching capacity is relative small, the MEC server can cache only a small part of segments, leading to a very low cache direct hit ratio. As a result, the majority of user requests ought to be responded to by

China Communications • July 2019

Authorized licensed use limited to: University of New South Wales. Downloaded on November 20,2020 at 02:28:37 UTC from IEEE Xplore. Restrictions apply.

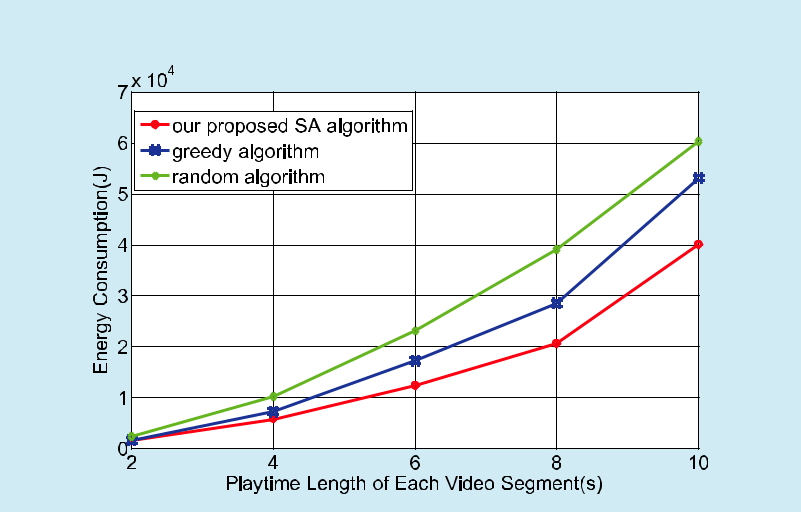
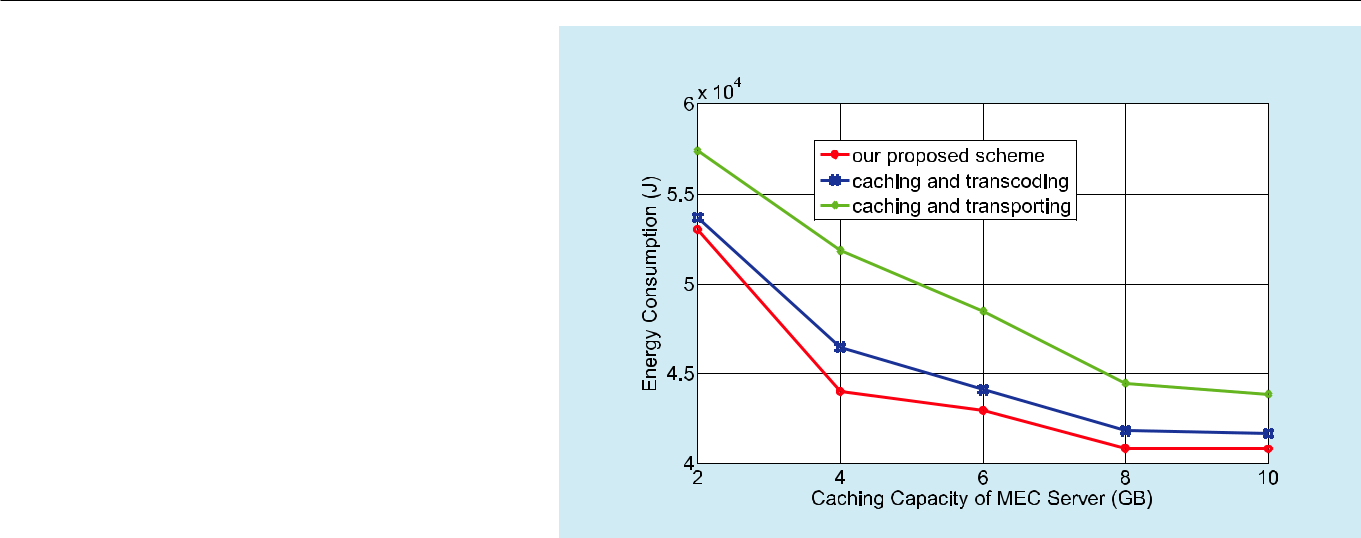
transcoding or transporting from backhaul, thus producing a significant energy consump-tion. Meanwhile, leveraging the analysis in Figure 6, our proposed transcoding schedule will perform best when caching space is too small, which is proved by experimental results in Figure 7. As caching capacity increasing, the cache direct hit ratio raises, resulting in an energy consumption decline because more and more requests will be served directly by MEC server without transcoding or transporting from backhaul. When caching space is abun-dant, MEC server tends to cache all the video segments, leading to similar profits of the three strategies.

To prove the advantage of our proposed scheme based on simulated annealing algo-rithm, we draw a performance comparison of proposed scheme based on SA algorithm with two another classical algorithms: greedy algorithm and random algorithm. Figure 8 investigates the relationship between playtime length and energy consumption while Figure 9 and Figure 10 present the impact of bitrate interval and segment number on energy con-sumption under the three fore-mentioned al-gorithms. With the increasing of the three key factors: playtime length, the bitrate interval and the number of segments, the curves take an ascending trend and our proposed scheme based on SA algorithm performs best among the three algorithms while greedy algorithm ranks second. Figure 11 shows the relationship between caching capacity of MEC server and energy consumption. With the caching capac-ity increasing, the red and the blue curves in Figure 11 shows a downward trend while the green curve presents no evident trend. Fur-thermore, as the below four figures shows, our proposed scheme based on SA algorithm performs best thanks to the capacity of avoid-ing being stuck in a local optimization and random algorithm yields worst results due to its uncertainty.

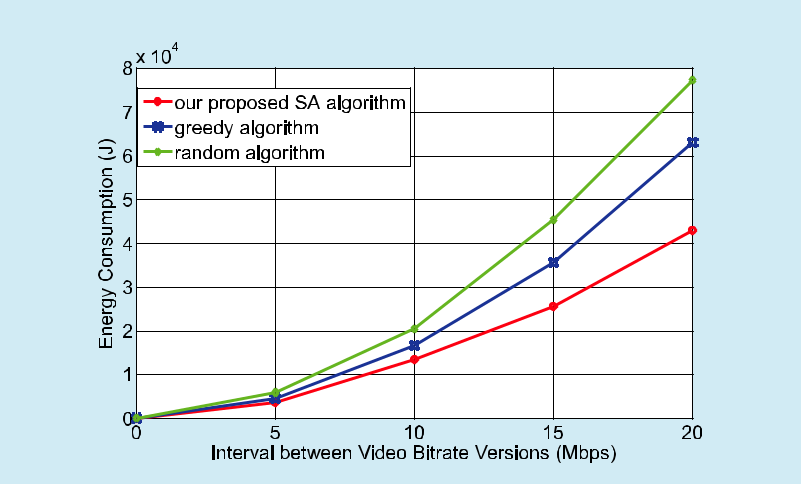
1. **CONCLUSIONS**

In this paper, we have proposed an energy ef-

**Fig. 7.** *Caching capacity of MEC server impact on energy consumption.*



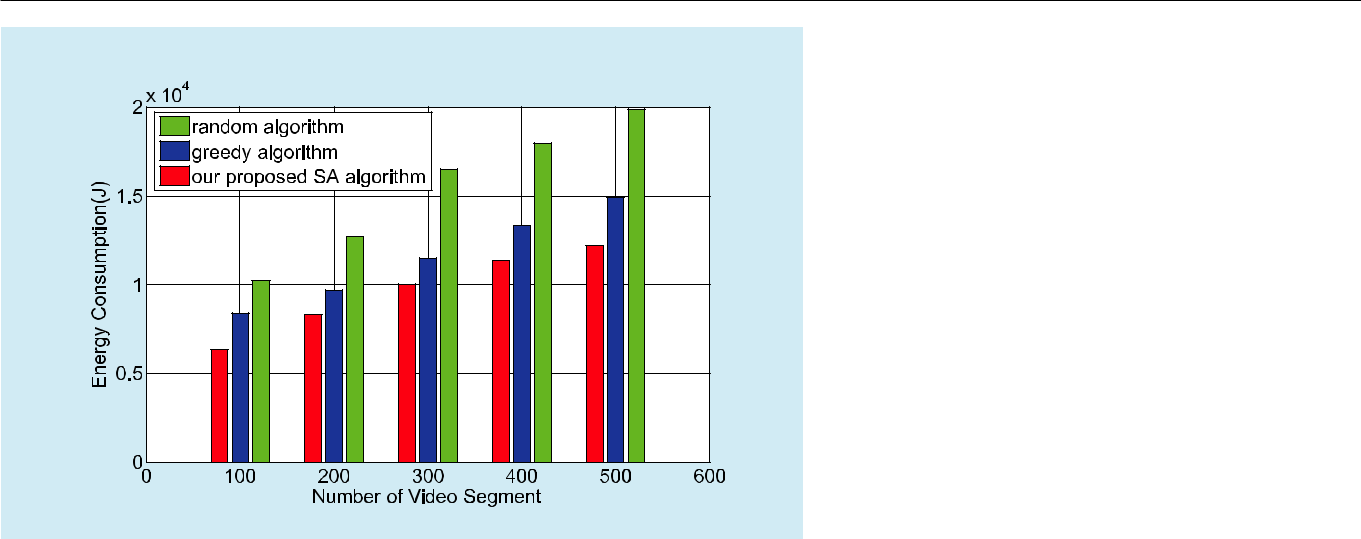
**Fig. 8.** *Playtime length of video segments impact on energy consumption.*



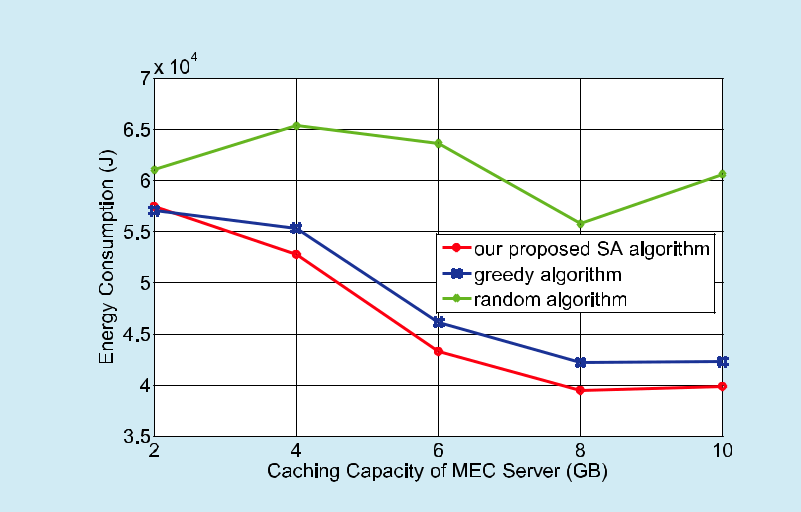
**Fig. 9.** *Interval between bitrate versions impact on energy consumption.*

|  |  |
| --- | --- |
| China Communications • July 2019 | 241 |

Authorized licensed use limited to: University of New South Wales. Downloaded on November 20,2020 at 02:28:37 UTC from IEEE Xplore. Restrictions apply.



**Fig. 10.** *Number of video segment impact on energy consumption.*



**Fig. 11.** *Caching capacity of MEC server impact on energy consumption.*

ficient joint caching and transcoding schedule strategy to dynamically determine the place-ment of requested video segments and the way of responding users’ content requests. We have formulated the problem as an integer program-ming problem to minimize the system energy consumption, thus realize energy-efficient vid-eo delivery in MEC-enabled cellular networks. To solve this problem, we have introduced a joint caching and transcoding scheme based on simulated annealing algorithm, which can iteratively approximate the globe best solu-tion. Simulation results have been illustrated to show that our strategy yields more signifi-cant gains, compared to another two strategies

without transcoding schedule. Experimental outcomes have also proved that our proposed scheme based on SA algorithm takes advan-tage of the ability of global optimization, and hence has better performance than classical algorithms such as greedy algorithm and ran-dom algorithm.

**ACKNOWLEDGEMENTS**

This work was support by the Major Na-tional Science and Technology Projects (No. 2018ZX03001014-003).

**References**

1. V. N. I. Cisco, “Forecast and methodology, 2017–2022,” *White Pape*r, 2018.
2. Y. Sani, A. Mauthe and C. Edwards, "Adaptive bitrate selection: a survey[J]," *IEEE Communi-cations Surveys & Tutorials*, vol. 19, no. 4, pp.2985-3014, Fourthquarter 2017.
3. H.Abouzeid, H.S.Hassanein, and S.Valentin, “En-ergy-efficient adaptive video transmission: ex-ploiting rate predictions in wireless networks[J],” *IEEE Transactions on Vehicular Technology*, vol.63, no. 5, pp. 2013-2026, Jun. 2014.
4. O. Oyman and S. Singh, “Quality of experience for HTTP adaptive streaming services[J],” *IEEE* *Communications Magazine*, vol. 50, no. 4, pp.

20–27, Apr. 2012.

1. X. Yin, A. Jindal, V. Sekar, and B. Sinopoli, “A control-theoretic approach for dynamic adap-tive video streaming over HTTP[J],” *ACM SIG-COMM Computer Communication Review*, vol.

45, no. 4, pp. 325–338, 2015.

1. J.Liu, R.Xie, and F.R.Yu, “Resource allocation and user association for HTTP adaptive streaming in heterogeneous cellular networks with small cells[J],” *China Communications*, vol. 13, no. 9, pp. 1–11, Sep. 2016.
2. R. Atawia, H. S. Hassanein, and A. Noureldin, “Robust long-term predictive adaptive video streaming under wireless network uncertain-ties[J],” *IEEE Transactions Wireless Communica-tions*, vol. 17, no. 2, pp. 1374–1388, 2018.
3. A. Argyriou, D. Kosmanos, and L. Tassiulas, “Joint time-domain resource partitioning, rate allo-cation, and video quality adaptation in hetero-geneous cellular networks[J],” *IEEE Transactions* *on Multimedia*, vol. 17, no. 5, pp. 736–745, May2015.
4. M. Patel, B. Naughton, C. Chan, N. Sprecher, S. Abeta, A. Neal, et al., “Mobile-edge comput-ing introductory technical white paper,” *White* *Paper, Mobile-edge Computing (MEC) Industry Initiative*, 2014.

242 China Communications • July 2019

Authorized licensed use limited to: University of New South Wales. Downloaded on November 20,2020 at 02:28:37 UTC from IEEE Xplore. Restrictions apply.

1. Y. C. Hu, M. Patel, D. Sabella, N. Sprecher, and V. Young, “Mobile edge computing: a key technol-ogy towards 5G,” *ETSI White Paper*, vol. 11, no. 11, pp. 1–16, 2015.
2. S. Wang, X. Zhang, Y. Zhang, L. Wang, J. Yang, and W. Wang, “A survey on mobile edge net-works: convergence of computing, caching and communications[J],” *IEEE Access*, vol. 5, pp. 6757–6779, 2017.
3. A. Mehrabi, M. Siekkinen, et al., “QoE-traffic optimization through collaborative edge cach-ing in adaptive mobile video streaming[J],” *IEEE* *Access*, vol. 6, pp. 52261–52276, 2018.
4. T. X. Tran, A. Hajisami, P. Pandey, and D. Pompili, “Collaborative mobile edge computing in 5G networks: New paradigms, scenarios, and chal-lenges[J],” *IEEE Communications Magazine*, vol. 55, no. 4, pp. 54-61, April 2017.
5. T. X. Tran, P. Pandey, A. Hajisami, and D. Pompi-li, “Collaborative multi-bitrate video caching and processing in mobile-edge computing networks[C],” *2017 13th Annual Conference on* *Wireless On-demand Network Systems and Ser-vices (WONS)*, Jackson, WY, 2017, pp. 165-172.
6. X. Xu, J. Liu, and X. Tao, “Mobile edge comput-ing enhanced adaptive bitrate video delivery with joint cache and radio resource alloca-tion[J],” *IEEE Access*, vol. 5, pp. 16406–16415, 2017.
7. C. Li, L. Toni, J. Zou, H. Xiong, and P. Frossard, “QoE-driven mobile edge caching placement for adaptive video streaming[J],” *IEEE Transac-tions on Multimedia*, vol. 20, no. 4, pp. 965–984,2018.
8. D. Wang, Y. Peng, X. Ma, W. Ding, H. Jiang, F. Chen, and J. Liu, “Adaptive wireless video streaming based on edge computing: opportu-nities and approaches[J],” *IEEE Transactions on* *Services Computing*, pp. 1–1, 2018.
9. Y. Chen, S. Zhang, S. Xu, and G. Y. Li, “Fun-damental trade-offs on green wireless net-works[J],” *IEEE Communications Magazine*, vol. 49, pp. 30–37, June 2011.
10. F. R. Yu, X. Zhang, and V. C. Leung, “Green com-munications and networking[B],” *CRC Press*,

2012.

1. Z. Li, R. Xie, Q. Jia, and T. Huang, “Energy-effi-cient joint caching and transcoding for HTTP adaptive streaming in 5G networks with mobile edge computing[C],” *2018 IEEE International* *Conference on Communications Workshops (ICC Workshops)*, Kansas City, MO, 2018, pp. 1-6.
2. B.Shen, S.-J.Lee, and S.Basu,“Caching strate-gies in transcoding-enabled proxy systems for streaming media distribution networks[J],” *IEEE* *Transactions on Multimedia*, vol. 6, no.2, pp.

375–386, April 2004.

1. J. Hachem, N. Karamchandani, and S. Diggavi, “Content caching and delivery over heteroge-neous wireless networks[C],” *2015 IEEE Confer-*

*ence on Computer Communications (INFOCOM),* Kowloon, 2015, pp. 756-764.

1. M. Savi, O. Ayoub, F. Musumeci, Z. Li, G. Vertica-le, and M. Tornatore, “Energy-efficient caching for video-on-demand in fixed-mobile conver-gent networks[C],” *2015 IEEE Online Conference* *on Green Communications (OnlineGreenComm)*,Piscataway, NJ, 2015, pp. 17-22.
2. R.Xie, Z. Li, T. Huang, and Y. Liu, “Energy-efficient joint content caching and small base station activation mechanism design in heterogeneous cellular networks[J],” *China Communications*, vol. 14, no.10, pp. 70–83, Oct. 2017.
3. P.Zhao, H.Tian, C.Qin,and G.Nie,“Energy-sav-ing offloading by jointly allocating radio and computational resources for mobile edge com-puting[J],” *IEEE Access*, vol. 5, pp. 11255–11268,

2017.

1. K. Zhang, Y. Mao, S. Leng, Q. Zhao, L. Li, X. Peng,
   1. Pan, S. Maharjan, and Y. Zhang, “Energy-effi-cient offloading for mobile edge computing in

5G heterogeneous networks[J],” *IEEE Access*, vol. 4, pp. 5896–5907, 2016.

1. P. J. Van Laarhoven and E. H. Aarts, “Simulated annealing[B],” *Simulated annealing: Theory and* *applications*, pp. 7–15, Springer, 1987.
2. A. Xenakis, F. Foukalas, G. Stamoulis, and T. Khattab, “Energy-aware joint power, packet and topology optimization by simulated annealing for WSNs[C],” *2013 7th IEEE GCC Conference* *and Exhibition (GCC)*, Doha, 2013, pp. 17-21.
3. S. Nahar, S. Sahni, and E. Shragowitz, “Simulated annealing and combinatorial optimization[C],” *23rd IEEE Conference on Design Automation*, pp.293–299, IEEE Press, 1986.
4. Y. Mao, J. Zhang, and K. B. Letaief, “Dynamic computation offloading for mobile edge com-puting with energy harvesting devices[J],” *IEEE* *Journal on Selected Areas in Communications*,vol. 34, no. 12, pp. 3590-3605, Dec. 2016.
5. A. Galanopoulos, G. Iosifidis, A. Argyriou, and
   1. Tassiulas, “Green video delivery in LTE-based heterogeneous cellular networks[C],” *2015* *IEEE 16th International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM),* Boston, MA, 2015, pp. 1-9.

**Biographies**

***Renchao Xie,*** received his Ph.D.



degree from the School of In-

formation and Communication

Engineering, BUPT, in 2012.

From July 2012 to September

2014, he worked as a postdoc-

toral researcher at China

Unicom. From November 2010 to November 2011, he visited Carleton University as a visiting scholar. He is an associate professor at BUPT. His current research interests include 5G net-

|  |  |
| --- | --- |
| China Communications • July 2019 | 243 |

Authorized licensed use limited to: University of New South Wales. Downloaded on November 20,2020 at 02:28:37 UTC from IEEE Xplore. Restrictions apply.

works and edge computing. He has published more than 50 journal and conference papers.

***Zishu Li,*** received his B.S. de-gree in communication engi-neering from Beijing University of Posts and Telecommunica-tions, she is now pursuing M.S. Degree in communication and information systems in Beijing University of Posts and Tele-



communications. Her research interests include 5G network, mobile edge computing and so on.

***Jun Wu,*** DHURAKIJ PUNDITUNIVERSITY’s MBA major, once employed by LEMKO in china for a long time as a distributed core network(software defined network)engineer, responsible for Design and Deploy Enter-prise private Network with mo-



bile communication system. From 2G to 5G, he has been committed to the evolution of virtualization network elements based on NFV, focus study solu-tions are based on flat network architecture. As a participant and leader of the project, he used the NFV-based integrated solutions of distributed core network elements and base station side to provide solutions for a series of projects such as UN peace-keeping and STATE GRID private network operation

and maintenance. Before that, he was also engaged in the optimization of mobile network for a long time and had strong theoretical and practical experience in the propagation characteristics of electromagnetic wave and wireless resource algorithm.

***Qingmin Jia,*** received his B.Sdegree in communication en-gineering from Qingdao Tech-nological University in 2014. He is currently working toward his Ph.D. degree at the State Key Laboratory of Networking and Switching Technology of Bei-



jing University of Posts and Telecommunications (BUPT). His current research interests include ICN, content caching, 5G network, and game theory.

***Tao Huang,*** received his B.S.degree in communication en-gineering from Nankai Univer-sity, Tianjin, China, in 2002, and his M.S. and Ph.D. degrees in communication and informa-tion systems from Beijing Uni-versity of Posts and Telecom-



munications in 2004 and 2007, respectively. He is currently a professor at Beijing University of Posts and Telecommunications. His current research inter-ests include network architecture and software-de-fined networking.

244 China Communications • July 2019

Authorized licensed use limited to: University of New South Wales. Downloaded on November 20,2020 at 02:28:37 UTC from IEEE Xplore. Restrictions apply.