

Research Article

In-Network Caching and Edge Computing-Based Live Broadcasting Optimization for Football Competitions

Zhigang Li 

Jilin Institute of Physical Education, Changchun 130022, China

Correspondence should be addressed to Zhigang Li; wcxf2000@sina.com

Received 11 November 2020; Accepted 25 November 2020; Published 12 December 2020

Academic Editor: Jianhui Lv

Copyright © 2020 Zhigang Li. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The business of football competitions is called the number one sport in the world, thanks to more than one billion people's attention. With the development of big convergence media, the live broadcasting of football competitions gradually becomes industrialization and commercialization, which has a direct relationship with economic growth. For the live broadcasting of football competitions, the users focus more on quality of experience, i.e., definition and instantaneity. In terms of such two metrics, the current live broadcasting schemes are difficult to cover them well. Therefore, this paper exploits the emerging in-network caching and edge computing technologies to optimize the live broadcasting of football competitions, shorten for IELB. At first, the live broadcasting optimization framework based on in-network caching and edge computing is presented. Then, the auction-based method is used to address the task scheduling problem in the edge computing. In addition, a video compression algorithm based on adaptive convolution kernel is introduced to accelerate the video transmission and guarantee users to obtain the contents of football competitions as quickly as possible. The proposed IELB has been verified based on the collected real football competitions dataset by evaluating response time, and the experimental results demonstrate that IELB is feasible and efficient.

1. Introduction

With the progress of the times, the industrialization and commercialization development of football has been rapidly promoted. In all outdoor sports, the football obtains the highest productive value, owns the biggest influence, and has the most widespread concern. According to the statistical data, the total output value of football per year can account for 43.5% of that of all sports, reaching 400 billion dollars and exceeding some developed countries and regions' GDP. Thus, the football is worthy of being called the number one sport in the world [1, 2]. Furthermore, according to the statistical data from FIFA, there are 1.6 million teams with more than 0.2 billion athletes worldwide playing the various football competitions by the end of 2019 [3]. Especially, with the rapid development of mobile Internet and big convergence media, the live broadcasting of football competitions [4] gradually becomes industrialization and commercialization, which has attracted many Internet companies to

develop the related businesses. In fact, for the live broadcasting of football competitions, the users pay more attention to quality of experience (QoE) [5], i.e., high definition and nice instantaneity, which needs that the whole process of live broadcasting guarantees the fluency. However, for the two evaluation metrics (i.e., high definition and nice instantaneity), the current live broadcasting schemes based on content delivery networks (CDN) [6–8] paradigm are very difficult to cover them well. In other words, it is necessary to explore the new networking paradigm to support the live broadcasting of football competitions with high definition and nice instantaneity.

Information centric networking (ICN) [9–11] is a new emerging and popular networking paradigm, which inherently owns one considerably important feature, i.e., in-network caching, that is to say, the ICN router also has the caching ability [12, 13] to store the contents that are related with the live broadcasting of football competitions. However, different from both the CDN server and traditional

router's buffer, the cache size of the ICN router is between them, and the ICN router can also be deployed at the edge nodes to satisfy the users' requests as many as possible. In other words, this paper uses the in-network caching feature of ICN to help the live broadcasting optimization of football competitions so as to obtain high definition and nice instantaneity. In fact, the usage of ICN has two obvious advantages. On the one hand, there is no need for fanfare to deploy the expensive CDN server. On the other hand, the ICN router can be regarded as a nice and big cache pool to accommodate more intermediate data stream. The two advantages can motivate the application of ICN into live broadcasting of football competitions.

As abovementioned, the ICN router can be deployed at the edge nodes which usually refer to the mobile devices, such as smartphones. Given this, the situation of mobile edge computing (MEC) [14–16] has to be considered and addressed. In fact, MEC has the widespread application value, especially for the optimization of live broadcasting. To be specific, the mobile devices have the limited storage resources and computation resources, and thus, it is very difficult for them to completely handle all tasks which are related to the live broadcasting. Therefore, it is one possible solution to offload some complex tasks at the edge server for computing, and the remaining simple tasks are performed at the local device. Under this condition, it is very important to address the task scheduling problem in the edge computing. At present, there have been some task scheduling methods [17–23], including artificial intelligence (AI) based ones [24, 25], but they usually cannot obtain the fast response speed and the low energy consumption. As a conclusion, it is necessary to explore the new method to address the task scheduling problem generated from the live broadcasting optimization.

Furthermore, the live broadcasting of football competition cannot do without the transmission of data stream. In other words, the video transmission is an indispensable operation to connect the contents provider and the users [26]. Regarding the video transmission, the video compression algorithm is very important, which can accelerate the video transmission and guarantee users to obtain the contents of football competitions as quickly as possible. However, the current video compression algorithms exit some limitations [27–31], and especially, the frame loss rate and the transmission time cannot reach the satisfactory level. Therefore, the video compression algorithm is also studied in this paper.

With the above consideration, this paper optimizes the live broadcasting of football competitions by using three aspects of technical points, i.e., the in-network caching feature of ICN, the task scheduling of edge computing, and the video compression of video transmission, called IELB. To sum up, the major contributions of this paper are concluded as follows. (i) The live broadcasting optimization framework based on in-network caching and edge computing is presented. (ii) The auction-based method is used to address the task scheduling problem in the edge computing. (iii) A video compression algorithm based on adaptive convolution kernel is introduced to accelerate the video transmission.

The rest of this paper is structured as follows. Section 2 reviews the related work. Section 3 presents the comprehensive system framework. Section 4 introduces the auction-based task scheduling. Section 5 proposes the video compression algorithm. Section 6 shows the significant experiment results. Section 7 concludes this paper.

2. Literature Review

2.1. Task Scheduling. There have been some research studies on task offloading proposals in the edge computing. In particular, some review papers have presented the comprehensive summary, such as [32–35]. Furthermore, in [17], the authors formulated successful computation probability, successful communication probability, and successful edge computing probability for offloading tasks to the MEC server. In addition, they also analyzed by simulation how the formulated probabilities vary for different sizes of task, task's target latency, and task arrival rate at the MEC server helping users to make offloading decision. In [18], the authors critically analyzed the resource-intensive nature of the latest existing computational offloading techniques for MEC and highlighted technical issues in the establishment of distributed application processing platforms at runtime where a prototype application was evaluated with different computation intensities in a real MEC environment. In [19], the authors presented a collaborative approach based on MEC and cloud computing that offloaded services to automobiles in vehicular networks. Meanwhile, a cloud-MEC collaborative computation offloading problem was formulated through jointly optimizing computation offloading decision and computation resource allocation. Besides, they also proposed a collaborative computation offloading and resource allocation optimization scheme and designed a distributed computation offloading and resource allocation algorithm to achieve the optimal solution. In [20], the authors proposed a price-based distributed method to manage the offloaded computation tasks from users. Therein, a Stackelberg game was formulated to model the interaction between the edge cloud and users so as to maximize the revenue subject to its finite computation capacity. For given prices, each user locally made offloading decision to minimize its own cost which was defined as latency plus payment. Depending on the edge cloud's knowledge of the network information, they developed the uniform and differentiated pricing algorithms, which could both be implemented in the distributed manner. In [21], the authors proposed a multi-user noncooperative computation offloading game to adjust the offloading probability of each vehicle in vehicular MEC networks and designed the payoff function considering the distance between the vehicle and MEC access point, application and communication model, and multivehicle competition for MEC resources. Also, they constructed a distributed best response algorithm based on the computation offloading game model to maximize the utility of each vehicle and demonstrated that the strategy could converge to a unique and stable equilibrium under certain conditions. In [22], the authors used the partial computation offloading

problem for multiuser in mobile edge computing environment with the multiwireless channel. The computation overhead model was built based on game theory. Then, the partial computation offloading algorithm with low time complexity was given to achieve the Nash equilibrium. In [23], the authors captured a user-centric view to tackle the offloading scheduling problem via jointly allocating communication and computation resources with consideration of the QoE of users where they formulated the design as a mix-integer nonlinear programming problem and solved it in an efficient way by the branch-and-bound method.

2.2. Video Compression. A number of video compression methods have been proposed, including some comprehensive survey papers, such as [36–38]. Furthermore, in [27], the authors maintained that a significant reduction in file size without sacrificing the visual quality could be achieved by using several efficient compression techniques. To this end, they proposed the video compression method by using global affine frame reconstruction, which involved affine parameter estimation for motion estimation and affine warping for motion compensation where the motion parameters were estimated and stored as compressed data. In [28], the authors proposed an adaptive transfer function based on perceptual quantizer for video compression, which used a fixed mapping curve from luminance to luma, i.e., the proposed transfer function adaptively mapped luminance to luma according to the contents. In [29], the authors introduced a hybrid spatially and temporally constrained content-adaptive tone mapping operator to convert the input high dynamic range video into a tone mapped video sequence, which was then encoded using the high efficiency video coding standard. In particular, the proposed tone mapped video simultaneously exploited intraframe spatial redundancies and preserved interframe temporal coherence of the tone mapped video sequence. In [30], the authors proposed a novel rate control scheme in H.264 to control the rate of compression ratio where the level of compression was decided with the help of the rate controller scheme. They measured the quality of the transmitted video and available bandwidth with the proposed technique and also built a quality multimedia content and transferred over the transmitter. In [31], the authors provided a lightweight video compression scheme through interframe and intraframe compression. In interframe compression, redundant frames were removed by a proposed interpolation search-based method and a lightweight edge detection technique. Then, intraframe compression was performed by a proposed adaptive column dropping technique modifying an existing technique. Besides, they also devised two reconstruction filters targeting to improve reconstruction quality.

3. System Framework

By using the in-network caching ability of ICN and edge computing, the proposed system framework of live broadcasting optimization of football competitions is shown in Figure 1. We can see that there are two kinds of different

servers, i.e., contents provider server used to store the football competitions related videos and edge computing server used to compute the complex tasks. The whole live broadcasting refers to the video transmission from the contents provider to the mobile device via some indeterminate ICN routers used to store the hot data stream. In particular, during the process of video transmission, a video compression algorithm based on adaptive convolution kernel is introduced to accelerate the video transmission. The live broadcasting contents arrive at the mobile device. Given the limited storage resources and computation resources, the complex tasks are scheduled to the edge computing server for computing, while the remaining simple tasks are performed at the local mobile device. Regarding this, the auction-based method is used to address the task scheduling problem in the edge computing.

In particular, the ICN router stays at the network level instead of the application level (e.g., CDN), which is assumed with the enough cache size to store the hot data stream. In addition, this paper also assumes that all mobile devices have the same configuration. In terms of two servers, the contents provider server has the very abundant space to store these videos of football competitions, just like a top-level distribution server. Differently, the edge computing server is only a high performance computing and storage server, and its computing ability and storage ability is not a circumstance to the contents provider server. The whole workflow of live broadcasting optimization of football competitions is shown in Figure 2. According to the above statements, task scheduling and video compression are two major research points in this paper, which will be addressed in the following sections.

4. Task Scheduling

The whole task scheduling in the edge computing includes two parts, i.e., offloading decision used to determine whether the tasks need to be offloaded and the scheduling method used to make resources computation and allocation.

4.1. Offloading Decision. Regarding whether the task is offloaded, its decision depends on whether the task's running time and energy consumption can be decreased in case of performing task offloading. Given this, two conditions, i.e., local performing and offloading performing, are considered and analyzed.

At first, for the arbitrary task τ_i , when it is performed at the local mobile device, the required time is defined as follows.

$$t_l = \frac{C_i}{v_l}, \quad (1)$$

where c_i is the required computation resources which refer to the number of CPU cycles, and v_l is the execution rate of CPU. Furthermore, the required energy consumption is defined as follows.

$$e_l = t_l p_l, \quad (2)$$

where p_l is the power of the mobile device.

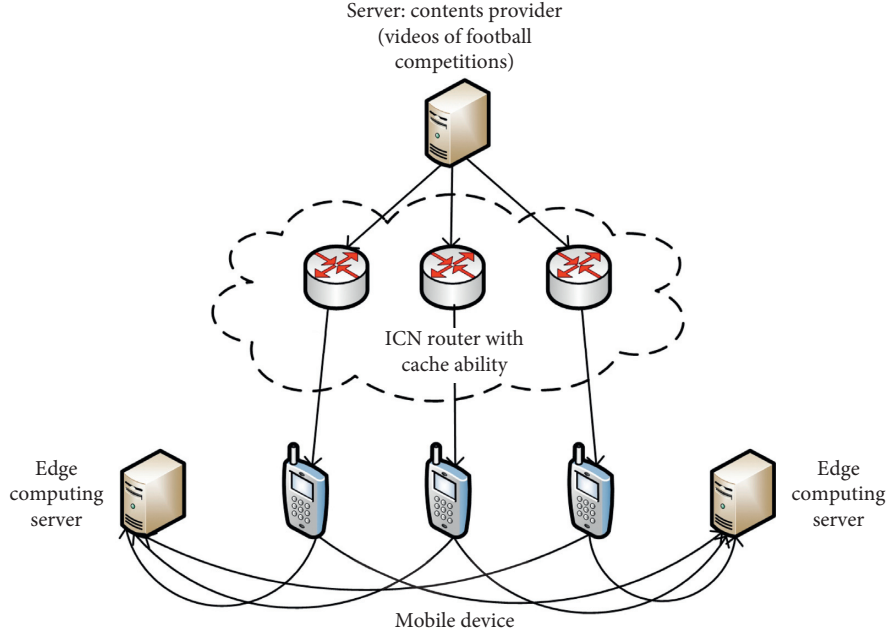


FIGURE 1: The proposed system framework of live broadcasting optimization.

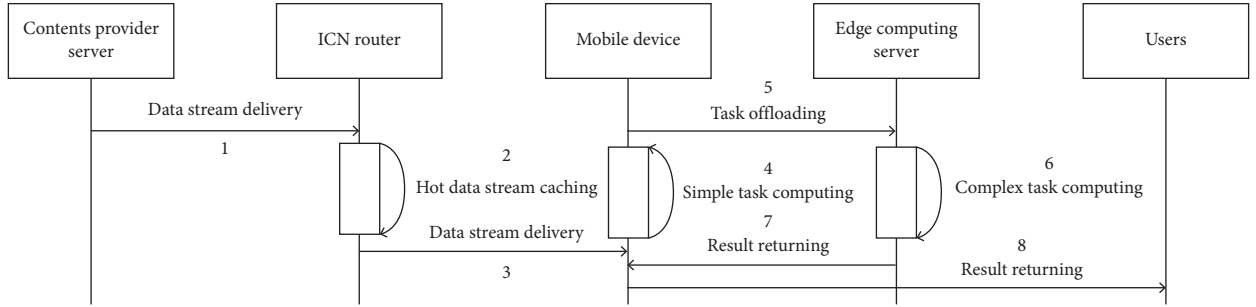


FIGURE 2: The workflow of IELB.

Then, when task_{*i*} is performed at the edge computing server, the required time t_{off} is composed of three parts, i.e., transmission time of task_{*i*} from the mobile device to the edge computing server, computing time at the edge computing server, and returning time of computing result from the edge computing server to the mobile device, denoted by t_{up} , t_{down} , and t_e , respectively. Mathematically,

$$t_{\text{off}} = t_{\text{up}} + t_e + t_{\text{down}}. \quad (3)$$

Consider that the size of the output result is far smaller than that of input data for most applications, and the returning time has no the significant influence on the total required time. On this basis, the above equation (3) is modified as follows.

$$t_{\text{off}} = t_{\text{up}} + t_e,$$

$$t_{\text{up}} = \frac{D_i}{W \log_2(1 + (p_l \text{Los}/N))}, \quad (4)$$

$$t_e = \frac{C_i}{v_e}.$$

Among them, D_i is the size of task_{*i*}, N is the Gaussian noise power of channel, W is the bandwidth of channel, Los is the transmission gain, and v_e is the edge computing server CPU's execution rate. To sum up, the total required time under such condition is expressed as follows.

$$t_{\text{off}} = \frac{D_i}{W \log_2(1 + (p_l \text{Los}/N))} + \frac{C_i}{v_e}. \quad (5)$$

Let e_{off} denote the total required energy consumption under such condition, as follows.

$$e_{\text{off}} = \frac{p_l D_i}{W \log_2 (1 + (p_l \text{Los}/N))}. \quad (6)$$

Moreover, when the consumed time and energy under the task offloading condition are smaller than those under the local performing condition, the task should be performed in the offloading way. For this purpose, the following two constraint conditions should be satisfied.

$$\begin{aligned} t_{\text{off}} &< t_l, \\ e_{\text{off}} &< e_l. \end{aligned} \quad (7)$$

For these two constraint conditions, the further derivation results are shown as follows.

$$v_e > \frac{v_l C_i}{C_i - ((D_i v_l) / (W \log_2 (1 + (p_l \text{Los}/N))))}, \quad (8)$$

$$W > \frac{v_l p_l D_i}{p_l C_i \log_2 (1 + (p_l \text{Los}/N))}, \quad (9)$$

which indicates that, if the computing ability of edge computing server satisfies in equation (8) and the current network bandwidth environment satisfies in equation (9), the task will be performed in the offloading way.

4.2. Scheduling Method. This paper uses the auction method [39] to address the scheduling strategy, including two roles, i.e., buyers and sellers. In terms of n buyers, they refer to these tasks, i.e., the set of buyers is denoted by $\text{TASK} = \{\text{task}_1, \text{task}_2, \dots, \text{task}_n\}$. Each task has two attributes, i.e., $\text{task}_i(t)$ and $\text{task}_i(p)$ which denote the required time to complete task_i and the bid of task_i , respectively. In terms of m sellers, they refer to the unoccupied virtual machines (VMs), i.e., the set of sellers is denoted by $\text{VM} = \{\text{vm}_1, \text{vm}_2, \dots, \text{vm}_m\}$. Each VM is equipped with one attribute, i.e., $\text{vm}_i(t)$, which denotes the rental time of vm_i . Furthermore, let $\text{price}_{\text{ini}}$ denote the initial bid price, and it is defined as follows.

$$\text{price}_{\text{ini}} = \frac{1}{t_l - t_{\text{off}}}. \quad (10)$$

Although equation (10) considers the emergency degree of task, there exists the unfairness. Especially, the task in the offloading way will gradually lose the price advantage. Therefore, this paper proposes the compensation strategy in terms of the condition of bidding failure. Mathematically, equation (10) is modified as follows.

$$\text{price} = \frac{1}{t_l - t_{\text{off}} - s N_{\text{fai}}}, \quad (11)$$

where s is the length of time slice, and N_{fai} is the number of bidding failures.

The derivation operation in terms of s is performed, and the following equation is obtained.

$$\frac{\partial \text{price}}{\partial s} = \frac{s}{(t_l - t_{\text{off}} - s N_{\text{fai}})^2}, \quad (12)$$

which indicates that the derivation value increases with the increase of N_{fai} . In other words, the large derivation value means the fast increasing speed in terms of price. Therefore, another meaning of equation (12) is that, more price compensations are performed based on more bidding failures.

Besides task and VM, there is a set of intermediate variables (denoted by CA) used to store some tasks satisfying auction condition. If $\text{task}_i(t) < \text{vm}_i(t)$, it means that task_i satisfies the auction condition, and it is added into CA. Otherwise, task_i is marked as the bidding failure. Regarding the whole auction process, reference [39] has presented the detailed steps, and this paper does give the corresponding steps no longer. In particular, the time complexity is $O(m \log m)$ rather than $O(nm)$.

5. Video Compression

The whole video compression process depends on the motion compensation which is used to realize the motion evaluation among video frames and pixel information compensation. In particular, the part is completed based on adaptive convolution kernel [40]. Inspired by Simon et al. [40], this paper learns the motion offset in terms of the continuous video frames so as to realize the motion evaluation among video frames. Regarding such process, it is defined as follows.

$$I'_{c2} = f(I_{c1}, I_{c2}) = I_{c1} + \Delta_{i,i+1}. \quad (13)$$

Among them, I_{ci} denotes the frame i , $\Delta_{i,i+1}$ denotes the offset from frames i to $i+1$, I'_{ci} denotes the motion compensation on I_{ci} , and f is the mapping function when the adaptive convolution kernel is used.

When the adaptive convolution kernel is used, it needs two convolution kernels to do the motion prediction, denoted by $K_h(x, y)$ and $K_v(x, y)$, which are considered as the horizontal vector and the vertical vector of $K(x, y)$, respectively. Mathematically,

$$K(x, y) = K_h(x, y) * K_v(x, y), \quad (14)$$

where $*$ denotes the convolution operation. In particular, the rectified linear units function is added after each convolution operation so as to enhance the express ability of the network. On this basis, the general motion compensation on I_{ci} is defined as follows.

$$I'_{ci}(x, y) = K_h(x, y) * K_v(x, y) * I_{ci}(x, y). \quad (15)$$

In summary, by using the network self-learning operation, the motion compensation is analyzed and performed based on $K_h(x, y)$ and $K_v(x, y)$ with the nonlinear mapping relationship. In terms of $K(x, y)$ with $K \times K$ convolution kernels, the current number of parameters is $2K$ instead of K^2 , which indicates that two one-dimensional convolution kernels have better computation efficiency than one two-dimensional convolution kernel.

TABLE 1: The average transmission times (ms) for ATS, IoTJ, and CoMNeT.

Data size	10 GB	20 GB	30 GB	40 GB	50 GB	60 GB
ATS	26.351	29.462	35.662	43.506	54.102	67.389
IoTJ	35.118	37.525	42.727	52.354	64.178	75.065
CoMNeT	48.672	49.192	55.881	64.394	77.296	86.406

TABLE 2: The independent transmission times (ms) for ATS, IoTJ, and CoMNeT where the data size is 50 GB.

No.	1	2	3	4	5	6	7	8
ATS	54.069	53.945	54.261	53.886	54.612	54.119	53.687	54.234
IoTJ	61.598	65.363	68.552	64.364	62.506	62.395	63.094	65.556
CoMNeT	78.932	79.642	77.265	78.345	77.064	76.355	75.106	75.663

6. Experiment Results

This paper makes three parts of experiments. At first, the proposed auction-based task scheduling strategy called ATS is verified. Then, the proposed adaptive convolution kernel-based video compression scheme called ACK is verified. Finally, the proposed whole IELB is verified. Among them, the first part considers transmission time and energy consumption as the evaluation metrics where references [21, 22] are used as the baselines, shorten for IoTJ and CoMNeT, respectively. The second part considers frame loss rate and transmission time as the evaluation metrics where references [30, 31] are used as the baselines, shorten for CoMCoM and JVCIR, respectively. The third part considers response time and QoE of users as the evaluation metrics by collecting the 1000 football competitions and testing 500 users.

6.1. Experiments on Task Scheduling. The average transmission times under different data sizes (from 10 GB to 60 GB) for ATS, IoTJ, and CoMNeT are shown in Table 1. We can see that the proposed ATS task scheduling strategy in the edge computing has the smallest average transmission time, followed by IoTJ and CoMNeT. In addition, with the increase of data size, the corresponding average transmission time increases. It suggests that the auction-based task scheduling strategy is more efficient than IoTJ and CoMNeT. Furthermore, when the data size is 50 GB, the independent transmission times under different experiments (from 1 to 8) for ATS, IoTJ, and CoMNeT are shown in Table 2. It is obvious that ATS has the smallest transmission time for each experiment. In particular, for 8 experiment results in terms of the independent transmission time, the proposed ATS has the best stability, followed by CoMNeT and IoTJ. It suggests that the proposed task scheduling is the optimal.

Moreover, the average energy consumption under different data sizes (from 10 GB to 60 GB) for ATS, IoTJ, and CoMNeT are shown in Table 3. We can see that the proposed ATS consumes the smallest energy. In summary, ATS has the smallest transmission time and the smallest energy consumption, which indicates that the proposed task scheduling strategy in the edge computing is considerably satisfactory.

TABLE 3: The average energy consumption (kJ) for ATS, IoTJ, and CoMNeT.

Data size	10 GB	20 GB	30 GB	40 GB	50 GB	60 GB
ATS	50263	50652	51297	52096	53121	54465
IoTJ	68355	68626	69172	70104	71233	72531
CoMNeT	83623	84069	84707	85391	85426	86829

6.2. Experiments on Video Compression. The average frame loss rates under different resolution ratios (i.e., 320P, 480P, 720P, and 1080P) for ACK, CoMCoM, and JVCIR are shown in Table 4. We can see that the average frame loss rate of ACK is the lowest. Although when the resolution ratio is 320P, their average frame loss rates have no significant difference, and the change of ACK is the linear while CoMCoM and JVCIR show the index movement. It suggests that the proposed video compression scheme can guarantee the integrated video transmission. Furthermore, the average transmission time under different resolution ratios for ACK, CoMCoM, and JVCIR are shown in Table 5. We can see that the average transmission time of ACK from the contents provider server to the mobile device is the smallest. Similarly, when the resolution ratio is 320P, their average transmission times have no significant difference. In particular, we also see that the average transmission time generated from video transmission is larger than that generated from the task offloading.

6.3. Experiments on Live Broadcasting Optimization. When the data size is 60 GB and the resolution ratio is 720P, the independent response times under different experiments (from 1 to 10) for IELB are shown in Table 6. We can see that the response time is around 120 ms with the ms-level, which can be acceptable by users.

Furthermore, this paper evaluates QoE of users where the QoE is divided into five grades, i.e., very satisfactory, satisfactory, borderline, dissatisfaction, and very dissatisfaction. From two perspectives (i.e., users and football competition), the evaluation results on QoE of users are shown in Table 7. We can see that the satisfaction rate can reach 100%, which indicates that the proposed live broadcasting optimization of football competition is feasible and efficient.

TABLE 4: The average frame loss rates (%) for ACK, CoMCoM, and JVCIR.

Resolution ratio	320P	480P	720P	1080P
ACK	0.015	0.039	0.051	0.073
CoMCoM	0.017	0.062	0.103	0.218
JVCIR	0.018	0.071	0.135	0.267

TABLE 5: The average transmission times (ms) for ACK, CoMCoM, and JVCIR.

Resolution ratio	320P	480P	720P	1080P
ACK	36.517	40.268	47.516	53.992
CoMCoM	47.068	53.267	64.333	82.908
JVCIR	61.285	68.593	79.112	97.267

TABLE 6: The average independent response times (ms) for IELB.

No.	1	2	3	4	5	6	7	8	9	10
IELB	120.236	119.351	120.692	119.864	121.006	120.443	121.507	119.868	120.612	121.323

TABLE 7: The statistical results on QoE of users.

Grade	Very satisfactory	Satisfactory	Borderline	Dissatisfaction	Very dissatisfaction	Satisfaction rate (%)
#Users	496	4	0	0	0	100
#Videos	998	2	0	0	0	100

7. Conclusions

This paper optimizes the live broadcasting of football competitions by using three aspects of technical points, i.e., the in-network caching feature of ICN, the task scheduling of edge computing, and the video compression of video transmission. At first, the live broadcasting optimization framework based on in-network caching and edge computing is presented. Second, the auction-based method is used to address the task scheduling problem in the edge computing. Third, a video compression algorithm based on adaptive convolution kernel is introduced to accelerate the video transmission. For these proposed strategies, this paper makes three kinds of experiments: (i) the proposed auction-based task scheduling strategy is verified by testing transmission time and energy consumption; (ii) the proposed adaptive convolution kernel-based video compression scheme is verified by testing frame loss rate and transmission time; and (iii) the proposed whole IELB is verified by testing response time and QoE of users. The experimental results demonstrate that the proposed IELB is feasible and efficient.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

- [1] T. Dima, "The business model of European football club competitions," *Procedia Economics and Finance*, vol. 23, pp. 1245–1252, 2015.
- [2] S. J. Koopman and R. Lit, "Forecasting football match results in national league competitions using score-driven time series models," *International Journal of Forecasting*, vol. 35, no. 2, pp. 797–809, 2019.
- [3] J. M. Buldu, D. R. Antequera, and J. Aguirre, "The resumption of sports competitions after COVID-19 lockdown: the case of the Spanish football league, chaos," *Solitons & Fractals*, vol. 138, pp. 1–7, 2020.
- [4] F. Zhou, L. Chen, and Q. Su, "Understanding the impact of social distance on users' broadcasting intention on live streaming platforms: a lens of the challenge-hindrance stress perspective," *Telematics and Informatics*, vol. 41, pp. 46–54, 2019.
- [5] W.-H. Hsu and C.-H. Lo, "QoS/QoE mapping and adjustment model in the cloud-based multimedia infrastructure," *IEEE Systems Journal*, vol. 8, no. 1, pp. 247–255, 2014.
- [6] S. Cui, M. R. Asghar, and G. Russello, "Multi-CDN: towards privacy in content delivery networks," *IEEE Transactions on Dependable and Secure Computing*, vol. 17, no. 5, pp. 984–999, 2020.
- [7] A. Passarella, "A survey on content-centric technologies for the current internet: CDN and P2P solutions," *Computer Communications*, vol. 35, no. 1, pp. 1–32, 2011.
- [8] N. Anjum, D. Karamshuk, M. Shikh-Bahaei, and N. Sastry, "Survey on peer-assisted content delivery networks," *Computer Networks*, vol. 116, pp. 79–95, 2017.
- [9] A. V. Vasilakos, Z. Li, G. Simon, and W. You, "Information

- centric network: research challenges and opportunities,” *Journal of Network and Computer Applications*, vol. 52, pp. 1–10, 2015.
- [10] Y. Ren, J. Li, S. Shi, L. Li, G. Wang, and B. Zhang, “Congestion control in named data networking—a survey,” *Computer Communications*, vol. 86, pp. 1–11, 2016.
 - [11] J. Lv, X. Wang, and M. Huang, “ACO-inspired ICN routing mechanism with mobility support,” *Applied Soft Computing*, vol. 58, pp. 427–440, 2017.
 - [12] N. Lal, S. Kumar, G. Kadian, and V. K. Chaurasiya, “Caching methodologies in content centric networking (CCN): a survey,” *Computer Science Review*, vol. 31, pp. 39–50, 2019.
 - [13] M. Zhang, H. Luo, and H. Zhang, “A survey of caching mechanisms in information-centric networking,” *IEEE Communications Surveys & Tutorials*, vol. 17, no. 3, pp. 1473–1499, 2015.
 - [14] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, “A survey on mobile edge computing: the communication perspective,” *IEEE Communications Surveys & Tutorials*, vol. 19, no. 4, pp. 2322–2358, 2017.
 - [15] N. Abbas, Y. Zhang, A. Taherkordi, and T. Skeie, “Mobile edge computing: a survey,” *IEEE Internet of Things Journal*, vol. 5, no. 1, pp. 450–465, 2018.
 - [16] H. Elazhary, “Internet of things (IoT), mobile cloud, cloudlet, mobile IoT, IoT cloud, fog, mobile edge, and edge emerging computing paradigms: disambiguation and research directions,” *Journal of Network and Computer Applications*, vol. 128, pp. 105–140, 2019.
 - [17] B. Bahadur Bista, J. Wang, and T. Takata, “Probabilistic computation offloading for mobile edge computing in dynamic network environment,” *Internet of Things*, vol. 11, pp. 1–11, 2020.
 - [18] M. Shiraz, M. Sookhak, A. Gani, and S. A. A. Shah, “A study on the critical analysis of computational offloading frameworks for mobile cloud computing,” *Journal of Network and Computer Applications*, vol. 47, pp. 47–60, 2015.
 - [19] J. Zhao, Q. Li, Y. Gong, and K. Zhang, “Computation offloading and resource allocation for cloud assisted mobile edge computing in vehicular networks,” *IEEE Transactions on Vehicular Technology*, vol. 68, no. 8, pp. 7944–7956, 2019.
 - [20] M. Liu and Y. Liu, “Price-based distributed offloading for mobile-edge computing with computation capacity constraints,” *IEEE Wireless Communications Letters*, vol. 7, no. 3, pp. 420–423, 2018.
 - [21] Y. Wang, P. Lang, D. Tian et al., “A game-based computation offloading method in vehicular multiaccess edge computing networks,” *IEEE Internet of Things Journal*, vol. 7, no. 6, pp. 4987–4996, 2020.
 - [22] S. Zhou and J. Waqas, “The partial computation offloading strategy based on game theory for multi-user in mobile edge computing environment,” *Computer Networks*, vol. 178, pp. 1–12, 2020.
 - [23] J. Luo, X. Deng, H. Zhang, and H. Qi, “QoE-driven computation offloading for edge computing,” *Journal of Systems Architecture*, vol. 97, pp. 34–39, 2019.
 - [24] L. Ma, S. Cheng, and Y. Shi, “Enhancing learning efficiency of brain storm optimization via orthogonal learning design,” *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, p. 1, 2020.
 - [25] S. Cheng, L. Ma, H. Lu, X. Lei, and Y. Shi, “Evolutionary computation for solving search-based data analytics problems,” *Artificial Intelligence Review*, 2020.
 - [26] Y.-M. Hsiao, J.-F. Lee, J.-S. Chen, and Y.-S. Chu, “H.264 video transmissions over wireless networks: challenges and solutions,” *Computer Communications*, vol. 34, no. 14, pp. 1661–1672, 2011.
 - [27] D. R. J. Dolly, G. J. Bala, and J. D. Peter, “Performance enhanced spatial video compression using global affine frame reconstruction,” *Journal of Computational Science*, vol. 18, pp. 1–11, 2017.
 - [28] S. Yu and C. Jung, “Adaptive perceptual quantizer for high dynamic range video compression,” *Journal of Visual Communication and Image Representation*, vol. 58, pp. 25–36, 2019.
 - [29] C. Ozcinar, P. Lauga, G. Valenzise, and F. Dufaux, “Spatio-temporal constrained tone mapping operator for HDR video compression,” *Journal of Visual Communication and Image Representation*, vol. 55, pp. 166–178, 2018.
 - [30] K. Siva Kumar, S. Sasi Kumar, and N. Mohan Kumar, “Efficient video compression and improving quality of video in communication for computer endcoding applications,” *Computer Communications*, vol. 153, pp. 152–158, 2020.
 - [31] T. Pal and S. Das Bit, “Low overhead spatiotemporal video compression over smartphone based delay tolerant network,” *Journal of Visual Communication and Image Representation*, vol. 70, pp. 1–20, 2020.
 - [32] H. Lin, S. Zeadally, Z. Chen et al., “A survey on computation offloading modeling for edge computing,” *Journal of Network and Computer Applications*, vol. 169, pp. 1–25, 2020.
 - [33] Q.-H. Nguyen and F. Dressler, “A smartphone perspective on computation offloading—a survey,” *Computer Communications*, vol. 159, pp. 133–154, 2020.
 - [34] S. Ali, M. Ghobaei-Arani, and S. Ali, “A survey on the computation offloading approaches in mobile edge computing: a machine learning-based perspective,” *Computer Networks*, vol. 182, pp. 1–24, 2020.
 - [35] A. Bhattacharya and P. De, “A survey of adaptation techniques in computation offloading,” *Journal of Network and Computer Applications*, vol. 78, pp. 97–115, 2017.
 - [36] A. J. Tabatabai, R. S. Jasinschi, and T. Naveen, “Motion estimation methods for video compression—a review,” *Journal of the Franklin Institute*, vol. 335, no. 8, pp. 1411–1441, 1998.
 - [37] J.-S. Lee and T. Ebrahimi, “Perceptual video compression: a survey,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 6, no. 6, pp. 684–697, 2012.
 - [38] H. A. Choudhury, N. Sinha, and M. Saikia, “Nature inspired algorithms (NIA) for efficient video compression—a brief study,” *Engineering Science and Technology, an International Journal*, vol. 23, no. 3, pp. 507–526, 2020.
 - [39] I. Abraham, S. Athey, M. Babaioff, and M. D. Grubb, “Cookies: peaches, lemons, and cookies: designing auction markets with dispersed information,” *Games and Economic Behavior*, vol. 124, pp. 454–477, 2020.
 - [40] N. Simon, L. Mai, and F. Liu, “Video frame interpolation via adaptive convolution,” 2017, <https://arxiv.org/abs/1703.07514>.