Adaptive Streaming Transmission Optimization Method Based on Three-Dimensional Caching Architecture and Environment Awareness in High-Speed Rail

Jia Guo 1,2, Yexuan Zhu 1*, Jinqi Zhu 1, Fan Shen 1, Hui Gao 3 and Ye Tian 2

1 College of Computer and Information Engineering, Tianjin Normal University, Tianjin 300380, China; c04s316@bupt.cn (J.G.); 2311090048@stu.tjnu.edu.cn (F.S.)
2 The State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China; yetian@bupt.edu.cn
3 School of Computer Science (National Pilot Software Engineering School), Beijing University of Posts and Telecommunications, Beijing 100876, China; gaohui786@bupt.edu.cn

* Correspondence: zhuyexuan20010928@163.com

Abstract: In high-mobility scenarios, a user’s media experience is severely constrained by the difficulty of network channel prediction, the instability of network quality, and other problems caused by the user’s fast movement, frequent base station handovers, the Doppler effect, etc. To this end, this paper proposes a video adaptive transmission architecture based on three-dimensional caching. In the temporal dimension, video data are cached to different base stations, and in the spatial dimension video data are cached to base stations, high-speed trains, and clients, thus constructing a multilevel caching architecture based on spatio-temporal attributes. Then, this paper mathematically models the media stream transmission process and summarizes the optimization problems that need to be solved. To solve the optimization problem, this paper proposes three optimization algorithms, namely, the placement algorithm based on three-dimensional caching, the video content selection algorithm for caching, and the bitrate selection algorithm. Finally, this paper builds a simulation system, which shows that the scheme proposed in this paper is more suitable for high-speed mobile networks, with better and more stable performance.

Keywords: high-speed rail; cache; three-dimensional caching architecture; cross-layer optimization; streaming media

1. Introduction

With the development of network technology, the demand for streaming media is increasing. Video traffic accounts for a gradually increasing proportion of the total network traffic [1]. Nowadays, high-speed rail has become one of the most popular modes of intercity travel. In the high-speed rail network environment, users move fast, and frequent base station handovers cause frequent network channel quality jitter, frequent data retransmissions, increased media data transmission delay, and buffer overflow, which seriously reduce the user’s media experience.

Adaptive transmission technology can reduce the negative impact of network quality jitter on the user’s media experience [2]. Its method is to divide the media data into different bitrates, store them on the media server, and adaptively select the appropriate bitrate data for the user according to the changes in the network bandwidth. Common transmission protocols include Dynamic Adaptive Streaming over HTTP (DASH) and HTTP Live Streaming (HLS). Many scholars have studied the transmission of variable-bitrate videos. Feng et al. [3] studied the impact of video segment length on user experience, and proposed an adaptive variable-length video segment scheme, which makes dynamic decisions based on real-time network data and user information, and achieves a balance between the accuracy...
and overhead of variable-length segmentation. The machine learning prediction of the client segment request bitrate implemented at the network edge was studied in [4], and the client segment request prediction problem was formulated as a supervised learning problem, i.e., to predict the bitrate of the next segment request of the client, so as to prefetch it at the mobile edge and improve the user’s experience. The above research results reduce the impact of channel quality jitter on the user’s media experience, but are based on low-speed networks; and more constraints need to be considered in high-speed networks.

Cache placement can reduce video transmission delay and the negative impact of network bandwidth jitter. Cache technology is also one of the techniques commonly used by scholars to reduce video transmission delay. Xu et al. [5] proposed a user-assisted base station (BS) storage and cooperative prefetching scheme for high-speed railway (HSR) communication, which improves the system throughput. However, it does not consider the environmental factors of users watching videos. A mobile edge network streaming media optimization framework based on privacy-preserving learning was proposed in [6], which included a user data privacy scheme and dynamic video cache algorithm. However, the strategy proposed by the literature is in the scenario of low mobility speed, and does not consider some characteristics of the high-speed railway mobile scenario. In high-speed networks, users move very fast, so the calculation of cache placement location and time is also a very critical challenge.

This paper considers the characteristics of the high-speed railway network scenario, such as high-speed mobility and severe network channel quality jitter. Under the above constraints, this paper proposes an adaptive streaming media transmission method based on three-dimensional cache and environment awareness (TDCEA). The three dimensions referred to in this paper are the time dimension, the space dimension, and multilevel caching from server to user. In the time dimension, the multimedia data need to be placed on different base station servers according to time. The space dimension refers to caches located at different locations, such as base station, high-speed rail, and client. Multilevel caching refers to the multilayer cache architecture formed by different base stations, high-speed rails, and clients based on the spatio-temporal attributes.

According to the user’s channel quality and the environmental parameters of the user watching the video, the transmission of streaming media based on the three-dimensional cache architecture is mathematically modeled, the optimization goal of this paper is summarized, and the corresponding solution algorithm is proposed, aiming to reduce the transmission delay and improve the user’s QoE. The solution proposed in this paper needs to solve the following problems: The first is which data to store. On the high-speed rail, a large number of users watch videos, and it is a challenge to choose which media data to store. The second is how to choose the cache location (at the base station or on the high-speed rail). The third is when to store the data. The user moves at a high speed and the cache has a short validity period, so the cache placement time is also very critical. The fourth is how to choose the media bitrate so as to improve the user’s media experience.

The rest of the paper is organized as follows. Section 2 introduces the related research status of video cache technology and adaptive bit rate technology. Section 3 proposes the system architecture and system model. In Section 4, some algorithms to solve the problem are proposed. The simulation is given in Section 5. Section 6 analyzes the results of the experiments, and discussion in Section 7. Section 8 provides a conclusion.

2. Related Works

Scholars have carried out a great deal of research on cache technology in order to improve users’ media experience. In [7], Mamduhi et al. proposed a cross-layer controller for handling file transfer at different levels and applied it to wireless network transmission, thus better managing network resources and improving the network transmission efficiency. In [8], the evolution of 360-degree video view requests was predicted by a long short-term memory (LSTM) network, and the predicted content is prefetched to the temporary storage. To further improve the video quality of transmission, users located in the overlapping areas
of multiple small base stations are allowed to receive data from any of these small base stations, thereby enhancing the user’s video experience. With the development of edge computing technology, scholars have proposed streaming media transmission optimization strategies based on edge computing. In [9], the user’s QoE was used as the key indicator to drive the edge device offloading, and limited computing resources in were used to provide better service for the user. Nguyen et al. [10] dynamically allocated wireless resources to optimize the user’s QoE in LTE wireless cellular networks, used the cross-layer information of the network.

Setting memory in the streaming media transmission process is one of the effective method to reduce video transmission delay and improving user QoE. Many researchers have carried out a large amount of research on placement problems. The technical challenges and development directions of mobile edge cache were analyzed in [11]. In [12], Pham et al. stored media data on edge servers to reduce data transmission delay and proposed an algorithm to increase the data hit rate. Wang et al. [13] analyzed the current state of research on caching in edge computing and proposed a collaborative strategy for streaming media content in cloud-edge collaborative environment. For the cache data selection problem, researchers generally believe that storing popular content can minimize the energy consumption of data delivery. The caching data selection algorithm was studied by [14], by using small base stations to store the most popular files. In [15], Li et al. studied the relationship between multiple cache points and user service, and found that, in some cases, randomly caching files results in better performance than caching the most popular data.

Li et al. [16] studied the system modeling, large-scale optimization, and framework design of hierarchical edge caching in device-to-device assisted mobile networks. A layer-based non-orthogonal transmission scalable media cache technology was proposed in [17], which utilized the layer features in scalable video. In [18], Zhang et al. studied the challenges brought by the different levels of content popularity and different viewing demands of scalable video, and proposed different types of content corresponding transmission schemes. In [19], Tran et al. proposed a multirate cache (MRC) scheme for video offloading in dense device-to-device (D2D) networks. A scalable video coding collaborative cache architecture for drones and clients was proposed in [20], which provided high transmission rate content distribution and personalized viewing quality in hotspot areas. A reactive content cache strategy based on deep reinforcement learning was proposed in [21], which made decisions for new requested content. The main goal was to obtain higher cache gain with lower space cost. Xu et al. [22] proposed a new heterogeneous network video content secure edge cache scheme. In order to motivate the participation of edge devices, the Nash bargaining game was used to model the negotiation between content providers and edge storage, and the optimal request cache space of content providers and the optimal price of each edge device were analyzed together. In [23], Kim et al. studied the caching of part of the content, rather than the entire content, to enrich users’ QoE. And They proposed a cache strategy for data blocks of different quality. A 360° video cache solution based on viewpoint reconstruction was proposed in [24].

At present, many scholars have conducted in-depth research on the adaptive technology of streaming media. A novel AQM algorithm was proposed in [25], which was based on the hotspot technology of deep learning, and used a long short-term memory (LSTM) neural network to enhance the users’ perception of video quality. Meanwhile, the optimization of the buffer was also presented in [26]. Santos et al. designed a new rate-adaptive algorithm, which aimed to minimize the amount of rebuffering and maximize the video quality. Gupta et al. [27] proposed a data cache video adaptive bit-rate switching algorithm. The algorithm dynamically constructed a smooth factor based on the throughput and cache state, and performed exponential weighted averaging on the bandwidth to maintain the buffer balance and ensure the video quality.

The above literature studied the cache technology and variable-bitrate algorithm for streaming media, and proposed many valuable methods. However, the above research is based on low-speed mobile networks. In high-speed mobile networks, cache is not
only related to space, but also to time, and frequent base station handovers cause severe jitter in user channel quality; these factors all cause the performance of the above methods to decline. Therefore, this paper proposes an adaptive streaming media transmission optimization method based on three-dimensional cache and environment awareness.

The main contributions of this paper are as follows:

1. This paper proposes an adaptive streaming media transmission architecture based on three-dimensional cache and environment awareness, which considers the utility of the cache in time and space dimensions.
2. This paper digitizes the adaptive streaming media transmission scheme based on the three-dimensional cache and environment awareness, and summarizes the problems to be optimized.
3. This paper proposes an algorithm to solve the problems, which are the selection of video data, cache location, and time and bitrate selection, and verifies the performance of the proposed method via simulation.

3. Architecture and Model

The adaptive streaming media transmission architecture based on a three-dimensional cache and environment awareness proposed in this paper is shown in Figure 1: First, the base station collects the user’s media requests and analyzes the popularity of the videos according to the request files of a large number of users, and then stores these data in the sever of the base station or the high-speed rail. Then, the user selects the appropriate bitrate video according to the environmental parameters of watching the video, and first searches in the high-speed rail cache. If the required content is found, the current quality of the wireless channel does not need to be considered. If it is not found, the data request is reapplied according to the quality of the current network channel quality and is searched in the base station cache. If it is found, it is directly sent to the user. If it is not found, the data is resent from the server or transcode (if there are other bitrate data).

![Figure 1. The architecture interactive model proposed in this article.](image-url)

3.1. User’s QoE Model

Assume that the current time is \( t \), \( n \) base stations are passed along the way, and the video is evenly divided into \( m \) segments of length \( T_s \). In this paper, in the high-speed mobile scenario, the distance between the user equipment (UE) and the base station (BS), and between the rail speed and other viewing environment, as well as the objective quality of the video, have a great impact on the user’s QoE, which is defined by Equation (1):

\[
QoE = \alpha \times \sigma(QoS) - \beta \times \phi(t) - \Theta \times T_r - \iota R_{ij} - \rho N_{rt}, \tag{1}
\]

where the function \( \sigma \) represents the objective quality. The objective quality \( \sigma \) can be expressed by the quality evaluation parameters PSNR and SSIM [27]. The function \( \phi \) represents the impact of cache miss and retranscoding on the user’s QoE, and \( T_r \) represents the impact of data retransmission delay after cache miss on the user’s QoE. The parameter
$R_{Js}$ represents the jitter of video quality, and the parameter $N_{rt}$ represents the number of rebuffering instances. The parameters $\alpha$, $\beta$, $\Theta$, $\iota$, and $\rho$ represent the impact coefficients of each factor.

$$R_{Js} = ((|J - J'| + 1)S_x)$$

where $R_{Js}$ denotes a monotonically increasing function of $S_x$, where $S_x$ is the number of bitrate switches that user $x$ has experienced in the previous segments. $J'$ is the current bitrate of user. $J$ is the video bitrate index.

Normalize the value of QoE:

$$QoE_{0,1} = \frac{QoE - QoE_{min}}{QoE_{max} - QoE_{min}}.$$ 

3.2. Video Data Cache Model

Content popularity: Content popularity is used to describe the popularity of streaming media, reflecting the probability of customers requesting streaming media files at a certain time. In this paper, the content popularity of streaming media files can be calculated by Equation (4): Assume a streaming media file $M$

$$M \in \{m_i | i = 1, 2, 3, \ldots, N\},$$

where $m_i$ represents the $i$-th video segment in $M$. Assign $m_i$ with a popularity level number, and the probability of accessing $m_i$ is $p_i$, for all $1 \leq i \leq j \leq N$, $p_i \geq p_j$. Therefore, $p_i$ can be called the popularity of $m_i$ in the set $M$, and the popularity distribution in $M$ is:

$$P = \{p_i | i = 1, 2, 3, \ldots, N\}.$$

Starting from each time segment, the base station collects requests from users, and the cache content is divided into two types: (i) New content, i.e., video segments requests that have not been cached yet. (ii) New version, i.e., the new version of the current video segment requested. Content popularity is modeled as a Zipf distribution. In this distribution, content of a higher popularity may have a higher probability of being requested. According to [28], $p_i$ can be obtained as follows:

$$p_i = \frac{s}{\sum_{j=1}^{F} p_j} (1 \leq i \leq F),$$

where $s \geq 0$ reflects the skewness of the popularity distribution.

$$s_c = \begin{cases} 1 & \psi < p_i < \lambda \\ 0 & p_i < \lambda \end{cases}$$

where $s_c = 1$ means storing the video data in the base station cache, and $s_c = 0$ means storing the video data in the high-speed rail cache. The parameters $\psi$ and $\lambda$ represent the popularity thresholds, which can be defined by the user. When $s_c = 1$, the base station and the storage time of the cache data need to be calculated. Assume that the user connects to the base station $BS_t$ at time $t$, the current speed of the user is $V_t$, the distance to $BS_t$ is $dis_t$, and the distance to $BS_{t+1}$ is $dis_{t+1}$. Therefore, the cache base station selection function is

$$BS_{\text{cache}} = \begin{cases} BS_t & dis_{t+1} - V_t < dis_t + V_t \\ BS_{t+1} & dis_{t+1} - V_t > dis_t + V_t \end{cases}.$$ 

Assuming that the base station cache is selected at $BS_{t+1}$, the cache validity can be calculated by Equation (9)

$$Timeliness = max \{t(i) | i = 1, \ldots, l, dis_{t+1} + V_t \times t(i) \leq dis_{t+2} - V_t \times t(i)\}.$$
3.3. Transmission Model

The objective quality of QoS can be calculated by Equation (10):

$$QoS = f(V_{\text{Bitrate}}), \quad (10)$$

where $f$ is the calculation function of video objective quality, and $V_{\text{Bitrate}}$ represents the bitrate selected.

$$V_{\text{Bitrate}} \leq \text{bandwidth}_t, \quad (11)$$

When the user needs to obtain data from the base station, the video bitrate chosen by the user should be less than the current network bandwidth $\text{bandwidth}_t$. The current network bandwidth is related to the user’s channel quality.

$$\text{bandwidth}_t = r(SINR), \quad (12)$$

Due to multipath effects, the signal from the base station reaches the user through $M$ different paths; therefore, the average signal-to-interference-plus-noise (SINR) ratio of a single user at a given point is expressed as [29]:

$$SINR(m) = \frac{\sum_{i=1}^{N} \sum_{p=1}^{M} \frac{w(\tau_i(m)+\delta_j)p_p}{q_i(m)}}{\sum_{i=1}^{N} \sum_{p=1}^{M} \frac{1-w(\tau_i(m)+\delta_j)p_p}{q_i(m)}} + N_0, \quad (13)$$

with

$$w(\tau) = \begin{cases} 1, & 0 \leq \tau \leq T_{cp} \\ 1 - \frac{\tau - T_{cp}}{T_u}, & T_{cp} \leq \tau < T_{cp} + T_u \\ 0, & \text{otherwise} \end{cases}, \quad (14)$$

where $M$ is the number of multipaths, $\tau_i(m)$ is the propagation delay with base station $i$ (for base station 0, $\tau_0(m) = 0$), $\delta_j$ is the additional delay that is added by path $j$, $q_i(m) = d_i^\alpha 10^{\xi_i}$ is the path loss from base station $i$, $\alpha$ is a constant and $\xi_i$ a lognormal variable due to shadowing. $P_p$ is the average power that is associated with the $j$-th path, $T_{cp}$ is the length of the cyclic prefix (CP), and $T_u$ is the length of the useful signal frame, and $N_0$ is the noise power.

The value of $SINR$ is related to the base station that the user connects to.

$$SINR = \gamma(C_{BS}), \quad (15)$$

where $\gamma$ is a function, and $C_{BS}$ is the base station to which that the user connects. The base station $C_{BS}$ is related to the current viewing environment of the user, such as the user’s moving speed, the distance between the user and different base stations, the size of video data transmission.

$$C_{BS} = w(V_t, dis_t, dis_{t+1}, V_{\text{Bitrate}}), \quad (16)$$

where $w$ is a function, $V_t$ is the user’s moving speed at time $t$, $dis_t$ is the distance between the user and the base station $BS_t$, $dis_{t+1}$ is the distance between the user and the base station $BS_{t+1}$, and $V_{\text{Bitrate}}$ is the bitrate. When the user needs to obtain data from the high-speed rail cache, the bitrate is independent of the current network condition of the user.

$$V_{\text{Bitrate}} = V_{\text{Bitrate}}', \quad (17)$$

where $V_{\text{Bitrate}}'$ represents the data bitrate in the high-speed rail cache. When the video data requested by the user is not found in the base station, there are two choices: request the video data with the corresponding bitrate from the server, or transcode the media data based on the computing module of the base station. If transcoding is chosen, there is an additional cost, which has a negative impact on the user’s QoE.
cost = V_{Bitrate} \cdot \frac{t_s}{\delta}, \quad (18)

where \( t_s \) represents the playback time of a video segment, and \( \delta \) represents the processor frequency of the edge service. The higher the frequency, the lower the cost. If the user chooses to retransmit the data, the data retransmission will cause delay \( T_r \).

\[ T_r = V_{Bitrate} \cdot \frac{t_s}{\text{bandwidth}_t}, \quad (19) \]

The optimization goal of this paper is

\[
\begin{align*}
\text{maximize} & \quad (QoE) \\
\text{Subject to} & \quad \text{Equation (1)} - (19).
\end{align*}
\quad (20)
\]

4. Algorithm
4.1. Three-Dimensional Cache Placement

In this paper, the base station collects requests from user devices in periods and processes the corresponding data requests in parallel. According to the popularity of the media, some data will be cached in the base station. When the user sends a request for \( m_i \), it will first search the high-speed rail \( B_{uk} \). If \( B_{uk} \) has the requested content, it will directly call \( m_i \) from \( B_{uk} \). If it does not, the BS will search for \( m_i \) in \( B_K \). If found, it will directly call \( m_i \) from \( B_K \). If none of the above is found, the content will be obtained from the video server. Algorithm 1 shows the three-dimensional cache access strategy. For \( BK \) and \( B_{uk} \), when receiving the content \( m_i \), if there is content \( m_i \) in the old version data, it will replace the old version with the new content. If the user has extra storage space, the content \( m_i \) will be cached directly. If both storage spaces are full, the content in \( B_{uk} \) is deleted, and then the content in \( B_K \) is deleted.

**Algorithm 1: Three-Dimensional Caching Access Policy (TDCAP)**

**Require:**
- System Initialization.

1: for \( t = 1 \) to \( T-1 \) do
2: Construct \( m_i(t) \) which obey Zipf. BS collects \( m_i(t) \).
3: for \( i = 1 \) to \( U \) do
4: newVersion = \( m_i(t) \cap m_i(1) \)
5: newContent = \( m_i(t) \cap \text{newVersion} \)
6: for \( nv \) in newVersion do
7: if \( nv \in B_{uk} \) then
8: Get content \( m_i(t) \) in \( B_{uk} \)
9: \( t = t + 1 \)
10: continue
11: end if
12: if \( nv \in B_K \) then
13: Get content \( m_i(t) \) in \( B_K \)
14: \( t = t + 1 \)
15: continue
16: else
17: get content \( m_i(t) \) from VS.
18: end if
19: end for
20: for \( nv \) in newContent do
21: Get content \( m_i(t) \) from media server VS
22: \( t = t + 1 \)
23: end for
24: end for
25: end for
4.2. Cache Content Selection

Pre-store content selection can be regarded as a multiple knapsack problem. The cache is regarded as a knapsack, and \( b \) bytes of space are allocated to UE\( k \) in \( B_k \) (the total capacity allocated to \( N \) users is less than or equal to the total capacity), so the capacity of is \( bB \). Similarly, the capacity of high-speed rail is \( B_{uk} \). There are \( n \) \( m_i \) that need to be downloaded to the cache, each \( m_i \) occupies space, and the popularity \( p_i \) of \( m_i \) is the value of \( m_i \). The specific content selection strategy is shown in Algorithm 2. The algorithm can achieve the local optimal solution of the cache strategy, and compared with other algorithms, it can significantly reduce the delay caused by the user not finding the target file.

**Algorithm 2:** Cache Content Selection Download Algorithm (CCSDA)

<table>
<thead>
<tr>
<th>Require:</th>
</tr>
</thead>
<tbody>
<tr>
<td>System Initialization ((dp[i][j])) representative the highest value maximum value of the first ( i ) items, given that the knapsack capacity is ( j ).</td>
</tr>
<tr>
<td>1: while ( bandwidth \geq \text{signal}_i ) do</td>
</tr>
<tr>
<td>2: for ( i = 1 ) to (</td>
</tr>
<tr>
<td>3: for ( j = 0 ) to ( bandwidth ) do</td>
</tr>
<tr>
<td>4: ( dp[i][j] = dp[i - 1][j] )</td>
</tr>
<tr>
<td>5: if ( j \geq v_i ) then</td>
</tr>
<tr>
<td>6: ( dp[i][j] = \max(dp[i][j], dp[i - 1][j - v_i] + p_i) )</td>
</tr>
<tr>
<td>7: end if</td>
</tr>
<tr>
<td>8: end for</td>
</tr>
<tr>
<td>9: end for</td>
</tr>
<tr>
<td>10: end while</td>
</tr>
<tr>
<td>11: Download ( M ) from ( VS ) to ( B_k ) according to ( dp[i][j] ).</td>
</tr>
<tr>
<td>12: for the user request ( M | b_u = \max(b_u) ) do</td>
</tr>
<tr>
<td>13: for ( i = 1 ) to (</td>
</tr>
<tr>
<td>14: for ( j = 0 ) to ( bandwidth ) do</td>
</tr>
<tr>
<td>15: ( dp[i][j] = dp[i - 1][j] )</td>
</tr>
<tr>
<td>16: if ( j \geq v_i ) then</td>
</tr>
<tr>
<td>17: ( dp[i][j] = \max(dp[i][j], dp[i - 1][j - v_i] + p_i) )</td>
</tr>
<tr>
<td>18: end if</td>
</tr>
<tr>
<td>19: end for</td>
</tr>
<tr>
<td>20: end for</td>
</tr>
<tr>
<td>21: Download ( M ) from ( VS ) to ( B_{uk} ) according to ( dp[i][j] ).</td>
</tr>
<tr>
<td>22: end for</td>
</tr>
</tbody>
</table>

4.3. Bit Rate Adaptive Selection Algorithm

This paper adopts edge computing and cross-layer information for bit rate adaptive algorithm. It is assumed that the video segment has multiple fixed bitrates: \( V_{\text{bitrate}} = V_1, V_2, \ldots, V_h \). The value of Bc is calculated according to Equation (21)

\[
V_c = \frac{V_h + V_m}{2},
\]

where \( V_h \) represents the bitrate of the highest quality level, and \( V_m \) is initially set to \( V_1 \). Then, the user’s QoE is calculated respectively, at the bitrates of \( V_h \) and \( V_m \). If the bitrate of \( V_h \) is chosen, the user’s QoE is denoted as \( QoE_1 \), and if the bit rate of \( B_m \) is chosen, the user’s QoE is denoted as \( QoE_2 \). Compare the values of \( QoE_1 \) and \( QoE_2 \): if \( QoE_1 > = QoE_2 \), then \( V_c = V_m \); otherwise, set \( V_c = V_h \). Repeat the above steps until \( h = m \). Then, the user’s QoE when choosing the bitrate is compared with the user’s QoE when choosing the video data stored in the high-speed rail. If \( QoE_3 > QoE_4 \), the data with the bit rate \( V_c \) are chosen, and if \( QoE_3 < QoE_4 \), the video data are requested from the high-speed rail cache. The pseudocode of the algorithm is shown in Algorithm 3.
Algorithm 3: Bitrate Adaptive Selection Algorithm

Require:
- System Initialization bitrates $V_{\text{bitrate}} = [V_1, V_2, ..., V_h]$,
- initial bitrate in the high-speed buffer is $bit_{\text{high}}$.

$m = 1$, $h = \text{len}(V_{\text{bitrate}}) - 1$, $c = (h + m) \div 2$

while $h \neq m$ do
    Calculate $V_h$
    Calculate $V_c$
    if $QoE(V_h) \geq QoE(V_c)$ then
        $m = c$, $c = \frac{h + m}{2}$
    else
        $h = c$, $c = \frac{h + m}{2}$
    end if
end while

if $QoE(c) > QoE(bit_{\text{high}})$ then
    Download video with bitrate $V_c$
else
    Download video with bitrate $bit_{\text{high}}$
end if

5. Simulation

In this section, we conducted a simulation experiment to verify the effectiveness of the proposed algorithm. First, we compared the cache performance, such as the average lifetime and the average hit probability of the cache data. Then, we compared the user’s QoE under the HLS, BSBC and BBA algorithms to verify the performance of the whole algorithm proposed in this paper.

Cache Performance Comparison Algorithm

In this paper, the average lifetime and the average hit probability of the cache data are defined as follows:

The average lifetime of the cache data:

This is used to describe the timeliness of the data collected by the system. The peak age of the content represents the maximum value of the content in the system; the age of the content is defined as follows:

$$A(t) = t - g(t), \quad (22)$$

where $t$ represents the current time, and $g(t)$ represents the generation time (also known as the generation time stamp) of the latest data packet received by the receiver. Therefore, the average information age is the sum of all instantaneous information ages divided by the amount of information:

$$\bar{A} = \frac{\sum_{t=1}^{N} A(t)}{N}, \quad (23)$$

where $N$ represents the total number of streaming media files.

Average hit probability:

The efficiency of cache is usually measured by the hit rate. In this paper, the hit rate refers to the probability that the file requested by the user is in the corresponding cache. The hit rate is defined as the ratio of the number of successful accesses to the total number of accesses. Specifically, it is the probability that a given access to a memory location will be satisfied by the cache. The hit rate can be expressed as follows:

$$H = \frac{N_h}{N_s}, \quad (24)$$

where $N_h$ represents the number of times that the requested file matches, and $N_s$ represents the total number of user requests. The average hit probability is defined as follows:
This paper uses Python to build a simulation environment, and the simulation parameters are shown in Table 1. The bandwidth data used in the simulation experiment are real bandwidth data collected in a real high-speed rail environment, and Figure 2 shows the network bandwidth in the high-speed rail environment used in this experiment. In order to compare the user’s QoE under different cache algorithms, the parameters in the QoE model are set as follows: $\alpha = 3.5$, $\beta = 0.15$, $\gamma = 1.5$, $\psi = 0.2$, $\rho = 0.2$. The transmission cost factor $\delta = 1.35$. In this paper, eight videos named Boat, FoodMarket, RitualDance, Crosswalk, TunnelFlag, Tango, Dancers, and DinnerScene are used as data for the streaming media experiment [30]. In the simulation experiment, each video is divided into 10 s segments, and the video is played in a loop, using the Joint Scaling video model (JSVM, version 9.19) [31] to encode the data with different quality levels.

Table 1. Simulation parameters.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Meaning</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$R$</td>
<td>Radius of $B_k$ signal range</td>
<td>240 m</td>
</tr>
<tr>
<td>$R'$</td>
<td>Radius of $B'_k$ signal range</td>
<td>12 m</td>
</tr>
<tr>
<td>$n$</td>
<td>Number of BS number of $B_k$</td>
<td>100</td>
</tr>
<tr>
<td>$n'$</td>
<td>Number of UE number of $B_{uk}$</td>
<td>120</td>
</tr>
<tr>
<td>$N$</td>
<td>Number of streaming media files</td>
<td>50 km</td>
</tr>
<tr>
<td>$s$</td>
<td>Zipf parameter of content</td>
<td>0.6</td>
</tr>
<tr>
<td>$b_B$</td>
<td>Size of individual cache on $B_k$</td>
<td>7200 MB</td>
</tr>
<tr>
<td>$b_u$</td>
<td>Size of individual cache on $B_{uk}$</td>
<td>60 MB</td>
</tr>
<tr>
<td>$T$</td>
<td>Total number of time slots</td>
<td>500</td>
</tr>
<tr>
<td>$V$</td>
<td>High-speed rail speed</td>
<td>350 km/h</td>
</tr>
<tr>
<td>$N_t$</td>
<td>Current bandwidth</td>
<td>Collected true value</td>
</tr>
</tbody>
</table>

Figure 2. Bandwidth data in high-speed rail 5G environment.

The following are the cache algorithms compared in this paper:

Content-Oriented Strategy: This strategy focuses on optimizing the content distribution in the streaming media system. The core idea of this strategy is to cache the requested content as close as possible to the user’s location in order to reduce latency and improve user experience.

Version-Oriented Strategy: Version-Oriented Strategy focuses on optimizing version management in the streaming media system. The core idea of this strategy is to cache different versions of the same content on different nodes to meet the needs of different users.

The following are the bitrate selection algorithms compared in this paper:

BSBC [32]: BSBC is the abbreviation of buffer-based video bitrate adaptation, which aims to solve the problem of low average bitrate in the DASH standard. The algorithm
analyzes the data in the video cache, calculates the average bitrate of the current frame, and dynamically adjusts the bitrate to improve the user experience. Specifically, the BSBC algorithm divides the video into multiple blocks, with each block containing multiple frames. For each block, the algorithm first calculates the total bitrate and total cache of the block, and then calculates the average bitrate of each frame according to these values. Finally, the bitrate is dynamically adjusted according to the average bitrate to improve the user experience.

HLS [33]: HTTP Live Streaming algorithm is a protocol and method for implementing the online streaming of audio and video content. This technology, developed by Apple, aims to use the HTTP protocol to transmit audio and video content efficiently and stably on the Internet, while automatically adjusting the bitrate to meet the needs of different network environments.

BBA [34]: The buffer-based adaptive (BBA) algorithm is a method of selecting bitrate based on buffer filling state. When the buffer is full, it chooses a higher bitrate; when the buffer is empty, it chooses a lower bitrate to avoid buffer overflow or playback interruption.

6. Results

Figure 3 shows the average lifetime of the cache data of the three algorithms as the number of files requested by $\text{UE}_k$ increases. As can be seen, as the total number of files increases, the average lifetime of the content decreases. This means that the files become increasingly fresh. The lower the freshness of the files, the higher the stability of the files obtained by the user, so our proposed algorithm has higher robustness.

![Figure 3. Average age of content with increasing number of files.](image)

As shown in Figure 4, as the number of file requests increases, the cache hit rate is higher because the cache content increases, so the overall QoE rises. However, the performance of the above two strategies is unstable, because their content requests are more random, and when the file is not found in the cache, the performance of the above two strategies will be decreased. Therefore, the strategy proposed in this paper has higher applicability in dealing with streaming media transmission in high-speed rail. Figure 5 shows the average hit probability of the three algorithms as the number of files requested by $\text{UE}_k$ increases. As the number of requested files increases, the files become increasingly difficult to hit. Our strategy ensures that the hit rate of the files is higher when the number of files is large.
As shown in Figure 6, as the number of users increases, the user’s QoE decreases because the cache space occupied by each user decreases. At this time, when the user requests a file, the probability of finding the requested file in the sever decreases, so the QoE decreases. However, the performance of the strategy proposed in this paper is still better than those of the above two strategies. The strategy proposed in this paper is more suitable for video transmission in high-speed mobile networks.

In Figure 7, the content movement of the video Boat is slow, so the encoded data are small, and the required bandwidth is small, resulting in the highest bitrate video playing smoothly for the user. The algorithm proposed in this paper can quickly find the local optimal solution, thus obtaining high QoE performance. In addition, in Figure 8, the QoE of the HLS algorithm is lower than that of other algorithms, especially at the beginning of the transmission, because the algorithm randomly selects the video bitrate at the beginning, making it more unstable. However, the algorithm proposed in this paper performs well in the initial bitrate selection. In Figure 9, watching the video RitualDance, because there are many elements in the video and the scene changes faster, the encoded data are larger, and thus users cannot watch the high bitrate video smoothly, resulting in lower QoE. However, the algorithm we proposed still maintains good performance, indicating that it can still perform well under high load. In general, the algorithm we proposed is relatively superior in high-speed environments.
Figure 6. QoE with increasing number of users.

Figure 7. The users’ QoE when watching the video Boat.

Figure 8. The users’ QoE when watching the video FoodMarket.
Figure 9. The users’ QoE when watching the video RitualDance.

Figure 10 shows the QoE of high-speed rail users when each user randomly selects one of eight videos to watch under different algorithms. The results show that the TDCEA method proposed in this paper achieves the best performance. Compared to high-speed rail users watching a single video individually, the average user QoE obtained in this experiment is lower than that of users watching the same video. This is because the increase in the number of videos available occupies more cache space, thus reducing the cache hit rate and the average user QoE.

The results in Figure 11 are obtained from simulations under the scenario where high-speed rail users randomly select one of eight videos to watch. In this paper, both the base station cache and the high-speed rail onboard cache vary with different levels. This shows the impact of different levels on the average user QoE. As illustrated, the proposed TDCEA method exhibits a significant increase in average user QoE with larger cache sizes. This suggests that expanding the onboard cache capacity in high-speed rail enables more video data to be cached, thereby reducing the demand for wireless resources and improving user video experience. The QoE for other methods also increases with larger cache levels, but to a lesser extent. This is because in high-speed rail scenarios, the bottleneck for video transmission lies between the base station and client side. Under the special networking environments of high-speed rails, it is difficult to store more data on the client side.
7. Discussion

In the high-speed rail network environment, the frequent switching of users causes the congestion algorithm to intervene frequently, resulting in a decline in the stability of the network bandwidth estimation accuracy. This paper proposes a new transmission method called TDCEA, which constructs a three-dimensional cache architecture to prestore media data at suitable locations and times. The simulation results demonstrate that our proposed TDCEA achieves the best performance. The main reason for this is that TDCEA has been designed specifically for high-mobility scenarios and is thus more suitable for such environments. As shown in Figure 11, TDCEA performs better with increased cache capacity because prefetching data in the high-speed rail environment improves the communication quality between the rail and clients. Moreover, the channel between the stationary high-speed rail environment and users is more stable. Therefore, clients can receive video data from the rail cache more smoothly. In summary, our proposed TDCEA delivers superior performance and robustness in highly mobile environments.

The method proposed in this paper requires cache servers to be deployed onboard high-speed rails. Also, due to the high speed of these rail systems, the timeliness of cached content is reduced, resulting in additional hardware costs. However, the trade-off between increased hardware expenses and improved user quality of experience has not been extensively discussed in this paper. This will be the focus of our future research by incorporating caching costs into our model to enable a more comprehensive cost-benefit analysis.

8. Conclusions

This paper proposes a new transmission method called TDCEA to address video delivery challenges in high-mobility environments. It solves the following three problems: one, when to prestore data; two, where to prestore data; and three, which data bitrate to select. Simulation experiments demonstrate that our proposed TDCEA delivers superior performance and robustness compared to existing methods under high-speed mobility scenarios.

Author Contributions: Conceptualization, J.G. and J.Z.; methodology, J.G.; software, Y.Z.; validation, Y.Z. and F.S.; formal analysis, Y.Z.; investigation, H.G.; resources, J.G.; data curation, Y.Z.; writing—original draft preparation, J.G.; writing—review and editing, J.G.; visualization, Y.Z.; supervision, Y.T.; project administration, Y.T.; funding acquisition, J.G., Y.Z., J.Z., F.S., H.G., and Y.T. All authors have read and agreed to the published version of the manuscript.

Funding: This work was substantially supported by the National Natural Science Foundation of China under Grant No. 62002263 and No. 62002025, in part by Tianjin Municipal Education Commission Research Program Project under 2022KJ012 and Open Foundation of State key Laboratory of
Networking and in part by Switching Technology (Beijing University of Posts and Telecommunications) (SKLNST-2021-1-20).

Data Availability Statement: The data presented in this study are available on request from the corresponding author. The data are not publicly available due to being related to our subsequent research.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations
QoE Quality of Experience
σ Objective Quality Function of Video
ϕ Negative Gain Function of QoE Due to Recoding
T The Negative Gain Function of QoE Due to Retransmission Delay
RJS The Negative Gain of QoE Due to Video Quality Jitter
Nrt The Negative Gain of QoE Due to Video Buffering Events
α, β, Θ, λ Trade-off Parameters
Sx The number of Bitrate Switches
x The x-th User
SINR Signal to Interference Plus Noise
r The Function for Computing SINR.
CBS Base Station for User Connection
VB bitrate The Bitrate of Cached Data in High-Speed Rail
Cost The Cost of Transcoding
ts The Duration of Playback for a Video Segment
δ CPU Performance
M Video File
N The Total Number of Video Segments
ψ, λ Popularity Threshold
Sc Boolean Function
Bs Base Station
t Time
Vt User Velocity
dist Distance Between User and Base Station
BS cache Cashed Base Station
VS Video Server
Timelines Cache Lifetime
f Objective Quality Function
VB bitrate Video Bitrate
bandwidtht Bandwidth
Tr Retransmission Delay
Bk Base Station Cache
Buk High-speed Rail Cache

References


Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.