Intelligent Caching for Mobile Video Streaming in Vehicular Networks with Deep Reinforcement Learning

Zhaohui Luo 1 and Minghui Liwang 2,*

1 Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences, Shenzhen 518055, China
2 School of Informatics, Xiamen University, Xiamen 361005, China

* Correspondence: minghuilw@xmu.edu.cn

Abstract: Caching-enabled multi-access edge computing (MEC) has attracted wide attention to support future intelligent vehicular networks, especially for delivering high-definition videos in the internet of vehicles with limited backhaul capacity. However, factors such as the constrained storage capacity of MEC servers and the mobility of vehicles pose challenges to caching reliability, particularly for supporting multiple bitrate video streaming caching while achieving considerable quality of experience (QoE). Motivated by the above challenges, in this paper, we propose an intelligent caching strategy that takes into account vehicle mobility, time-varying content popularity, and backhaul capability to improve the QoE of vehicle users effectively. First, based on the mobile video mean opinion score (MV-MOS), we designed an average download percentage (ADP) weighted QoE evaluation model. Then, the video content caching problem is formulated as a Markov decision process (MDP) to maximize the ADP weighted MV-MOS. Owing to the prior knowledge of video content popularity and channel state information that may not be available at the road side unit in practical scenarios, we propose a deep reinforcement learning (DRL)-based caching strategy to solve the problem while achieving a maximum ADP weighted MV-MOS. To accelerate its convergence speed, we further integrate the prioritized experience replay, dueling, and double deep Q-network technologies, which improve the performance of DRL algorithm. Numerical results demonstrate that the proposed DRL-based caching strategy significantly improves QoE, and achieves better video delivery reliability compared to existing non-learning approaches.

Keywords: multi-access edge computing; edge caching; mobile video; reinforcement learning

1. Introduction

The advent of the 5G era and rapid development of cellular vehicle-to-everything (C-V2X) technologies have realized a massive number of vehicles equipped with advanced intelligent devices (e.g., high-definition (HD) players, 3D navigation equipment, vehicle mixed reality (MR) glasses), which bring significant mobile data traffic and service explosion [1]. Furthermore, the global mobile data traffic is expected to reach 77 exabytes per month by 2022, as reported by Cisco, where 79 percent of this data comes from mobile video [2]. Additionally, innovative mobile video technologies have inspired a wide range of video streaming applications in vehicular networks, becoming one of the most popular and indispensable services.

Generally, a massive amount of HD mobile video data is downloaded through the backhaul network of a macro base station (MBS), resulting in congested traffic loads. Consequently, the limited backhaul bandwidth capacity poses one of the most significant challenges to HD video data delivery services in the internet of vehicles (IoV) [3]. To effectively alleviate the backhaul traffic load, reduce the content delivery latency, and meet the diverse quality of service (QoS) data transmission requirements [4], multi-access edge computing (MEC) has been proposed as an efficient paradigm [5], which enables data caching at the edge of mobile networks, thus offers fast, and video delivery. The quality of
experience (QoE) of mobile video streaming is a key criterion for measuring mobile video quality in mobile networks [6].

One promising method to improve the QoE of mobile users is to bring video content close to mobile users by deploying the MEC server (MECS) in mobile edge networks. Hence, mobile users can download the requested HD video from the small base stations (SBS) directly instead of acquiring the relevant contents from the MBS through the backhaul network, which reduces transmission delay for video distribution. However, the existing mobile network edge caching strategies can hardly be applied directly in vehicular edge caching networks for the following reasons [7,8]. First, high vehicle mobility and limited caching storage capability bring difficulties in achieving HD video delivery among vehicle users. Second, the high bitrate of the emerging Ultra HD videos (e.g., 1080p, 2K, and 4K video that request at least 12–25 Mbps user data rate [9]) may incur vehicle users to be unable to complete HD video download tasks within service hours of road side unit (RSU). Owing to the limited backhaul capacity, the MBS, which took over the service, can hardly provide bitrate HD video requirements, leading to failure to download tasks. Additionally, considering the high vehicle mobility, the video transmission time is limited by the time duration a vehicle is within the signal coverage of an RSU [10]. As a result, the traditional mobile network QoE evaluation method relying on available service time is infeasible to apply vehicular caching networks directly.

Adaptive bitrate (ABR) video streaming has recently become a technology to improve QoE [11]. However, many works have focused on the ABR video process, ignoring the heterogeneity of networks and user capabilities that impact ABR video caching. MECS can connect with vehicle users for high-speed and stable transmissions, but its storage capacity is limited [12]. Caching high-bitrate videos can achieve higher QoE than low-bit-rate videos, but requires higher caching resources. Hence, there is a trade-off between caching some high-bitrate videos and diverse videos [13,14]. Moreover, the duration for downloading HD video is limited by the constrained time that a vehicle passes through the RSU service area [15]. To achieve maximum QoE, the caching decision of MECS in vehicular networks becomes challenging to support mobile videos of multiple bitrate levels [16]. Motivated by the abovementioned challenges, in this paper, we propose a deep reinforcement learning (DRL)-based intelligent caching strategy for mobile video in vehicular networks. Major contributions are summarized as follows:

1. We propose a mobile video caching framework in vehicular networks, where vehicle users can access the high-resolution video from the MECS deployed on RSU directly without occupying the backhaul network to release the bandwidth resource and reduce the data transmission delay.
2. We designed an average download percentage (ADP) weighted mobile video mean opinion score (MV-MOS) model for vehicle users. Then, the video content caching problem is formulated as a Markov decision process (MDP) to maximize the ADP weighted MV-MOS.
3. Based on a deep Q-learning network (DQN), we propose an intelligent caching algorithm to solve the problem while achieving a maximum ADP weighted MV-MOS. Furthermore, a P3DVQC caching scheme in vehicular networks was proposed to improve the performance of algorithm by integrating the prioritized experience replay, dueling, and double DQN technologies.

The remainder of this paper is organized as follows. The related work is presented in Section 2. We describe the system model in Section 3. In Section 4, we formulate intelligent caching for mobile video problems. Section 5 gives the solution of the proposed method. Numerical results and discussion are presented in Section 6. Finally, a conclusion is drawn in Section 7.

2. Related Work

A considerable study has been conducted on the design of mobile edge caching algorithms for mobile networks. The authors in [17] proposed proactive caching of popular
content during off-peak periods to reduce peak traffic demands. The authors in [18] conducted a comprehensive survey regarding different aspects of mobile edge caching. Furthermore, they have discussed caching schemes based on different caching locations and performance criteria. The work in [19,20] leveraged edge nodes (e.g., small cells) to store popular content, such as multimedia files, to reduce latency and improve the performance of 5G networks. The authors in [21] propose a novel deep learning-based proactive caching framework in cellular networks that obtain higher backhaul offloading and user satisfaction. Along this line of mobile edge caching, video placement has been studied over different heterogeneity of networks. Considering that user devices and preferences and needs for specific videos may vary, adaptive bitrate (ABR) streaming becomes a pivotal technique to improve the quality of delivered video on networks. The authors in [22] envision a collaborative joint caching and processing strategy for multiple bitrate video delivered to adapt to the heterogeneity of user capabilities and wireless communication conditions. According to the authors in [23] scalable video coding (SVC) based video services are considered to formulate the joint video quality selection and caching problem, to maximize vehicular user’s QoE. The solutions based on SVC on a system called DASH to ensure the quality of streaming media services have been proposed in recent studies [24].

Many works focus on the optimization efficiency of video caching and reducing video delivery delay in vehicular networks. The work in [25] proposed a cooperative transmission strategy for video transmission in small-cell networks with caching. Authors in [26] investigate a problem of cooperative mobile edge caching for scalable video streaming in HetNets. Adaptive video technology is applied to popular streaming services such as YouTube, Netflix, and Youku to provide smooth streaming and improve quality, such as Microsoft smooth streaming, Adobe’s HDS, and Apple’s HLS [27]. These streaming services encode videos into multiple versions with discrete bitrates. The authors in [28] proposed a mechanism based on MEC to cache only the highest available bit-rate video content while converting it to the requested lower bit-rate version using the available processing power of MEC. Many recent works focus on an adaptive bitrate streaming to cope with time-varying channels incurred by vehicular users’ high mobility in IoV. In [29], the authors use a technique to effectively use both ABR streaming and BS caching in vehicular networks with high channel variations. Quality of experience serves as a direct evaluation of vehicle users’ experiences in mobile video transmission, and thus the authors in [30] propose a deep learning-based QoE prediction approach with a large-scale QoE dataset for mobile video transmission. The work [31] proposed to simultaneously optimize energy consumption and QoE metrics in video streaming over software-defined mobile networks (SDMN) combined with MEC.

Zhao et al. [32] by considering the interaction between video encoding and edge caching, the authors proposed a QoE-driven cross-layer optimization scheme for secure video transmission over the backhaul links in cloud-edge networks. Liang et al. [33] proposed enhancing the quality of experience-aware wireless edge caching with bandwidth provisioning in software-defined wireless networks. Latency is decreased, and the utilization of caches is improved in the proposed scheme. Huang et al. [34], based on the video popularity and the wireless resource conditions of the network, proposed a joint cache allocation and video delivery scheme for the video streaming system. Alberto et al. [35] presented demonstrates the possibility of developing a DRL-based quality optimization framework which can guarantee an adequate QoE. Li et al. [36] studied a QoE-driven mobile edge caching placement optimization problem for dynamic adaptive video streaming that by the optimal caching placement of representations for multiple videos, they maximize the aggregate average video distortion reduction of all users while minimizing the additional cost of downloading. Qiao et al. [37] proposed a deep deterministic policy gradient (DDPG)-based cooperative caching scheme to jointly optimize the content delivery and content placement in vehicular networks.
3. System Model

As shown in Figure 1, we consider a highway vehicular networks scenario that includes an MBS, several RSUs, and vehicle users (VUs). The MBS connects to the core network (CN) through the backhaul link. Denote \( M = \{1, \ldots, M\} \) and \( U = \{1, \ldots, U\} \) be the RSUs set and vehicle users set, respectively. Each RSU is equipped with an MECS of size \( \varphi_{MEC} \) to store a number of popular video replicas, which helps reduce the delay of content delivery and improve QoE of VUs. Vehicular users can access nearby RSU or MBS and download videos from the MECS or CN. We assume that each vehicular user request one of the interesting videos once the vehicle enters the service coverage areas of RSU but experiences a higher download rate if the requested one is pre-cached in MECS. Set the vehicle user arrival probability to a Poisson distribution of different parameters that obey the Markov process, then the parameter of vehicle user arrival probability at time slot \( \tau \) is recorded as \( \lambda(\tau) \in \{\lambda_1, \lambda_2, \ldots, \lambda_B\} \). Then, at time slot \( \tau \), the probability that the number of vehicle users \( U(\tau) \) in the service coverage is expressed as

\[
P(U(\tau)) = \frac{(\lambda(\tau)U(\tau))^U(\tau)!}{U(\tau)!} e^{-\lambda(\tau)}. \tag{1}
\]

![Figure 1. System model.](image)

We define the state transition probability of the parameter of arrival probability \( \lambda(\tau) = \lambda_i, i \in \{1, \ldots, B\} \) at time slot \( \tau \) transfers to \( \lambda(\tau + 1) = \lambda_j, j \in \{1, \ldots, B\} \) at time slot \( \tau + 1 \) as \( \vartheta_{i,j} \), the transition matrix \( \Gamma \) of parameter \( \lambda \) is expressed as

\[
\Gamma = \begin{pmatrix}
\vartheta_{1,1} & \cdots & \vartheta_{1,B} \\
\vdots & \ddots & \vdots \\
\vartheta_{B,1} & \cdots & \vartheta_{B,B}
\end{pmatrix}. \tag{2}
\]

3.1. Caching Model

The video content popularity reflects the statistical results of the video requested by vehicle users over a period. As a result, the time slot varying scale of video content popularity is much larger than the cache content update of the MECS. Furthermore, the time slot varying scale of the video content delivery process is much smaller than the cache content update of the MECS. Hence, the mobile video caching in the vehicular network can be modeled as a multi-time scale model, as shown in Figure 2.
Figure 2. Multi-time scale model.

Popularity variation time scale: We assume that one video popularity variation time slot is included $K_{CN}$ caching placement time slot. The time scales of popularity variation can be defined as $t^x$, $x \in \{1, 2, 3, \ldots, K_{CG}\}$, which containing $K_{CG}$ time slots, and define the length of the one MECS caching update period is $\Delta t^y$, then the length of $\Delta t^x$ is expressed as

$$\Delta t^x = K_{CN} \Delta t^y. \quad (3)$$

Caching placement time scale: The times scales of caching placement can be defined as $t^x_y$, $y \in \{1, 2, 3, \ldots, K_{CN}\}$, and caching video content of MECS is updated once every time slot of $t^x_y$. Suppose $D$ is the length of service covered, the length of $\Delta t^x_y$ at vehicles speed $v(t^x_y)$ is calculated as

$$\Delta t^x_y = \frac{D}{v(t^x_y)}. \quad (4)$$

Video delivery time scale: The rapid movement of the vehicle causes the geographic location and channel status to change, the transmission rate of the vehicle user is time-vary. The $\Delta t^x_y$ is discretized into $K_{SN}$ segments to simplify the model. Then the times scales of video delivery can be defined as $t^x_y_z$, $z \in \{1, 2, 3, \ldots, K_{SN}\}$. It can be considered that the communication rate in each content distribution period $\Delta t^x_y$ remains unchanged. The length $\Delta t^x_y_z$ of the time slot $t^x_y_z$ is calculated as

$$\Delta t^x_y_z = \frac{\Delta t^x_y}{K_{SN}}. \quad (5)$$

The video requested by vehicle users (such as 4K/8K high definition movies, etc.) comes from a video content library $F = \{f_1, f_2, \ldots, f_F\}$, which containing $F$ video files. Assumed that $F$ obeys Zipf’s law [27], and the videos of $F$ are arranged in descending order of their content popularity. Using constant bitrate (CBR) technology to encode the videos of $F$ into a constant bitrate multiple bitrate video $L = \{\ell_1, \ell_2, \ldots, \ell_L\}$, its bitrate level is still arranged in descending order $\ell_1 > \ell_2 > \ldots > \ell_L$, that is, $\ell_1$ is the replicas of video with the highest bitrate. Without loss of generality, assume that the time lengths of the videos in $F$ are equal, denoted as $\omega$, it can be considered that after coding using the CBR technology, all video copies of the same bitrate level in $F$ have the same size. The replicas size $g_j$ of video $f$ at the $\ell_j$ bitrate level is expressed as

$$g_j = \omega \ell_j. \quad (6)$$

Hence, the caching state matrix $X(t^x_y)$ of MECS at the time slot $t^x_y$ is defined as

$$X(t^x_y) = F \times L = \left\{ x_{f,\ell_j}(t^x_y), f_i \in F, \ell_j \in L \right\}_{F \times L}. \quad (7)$$
where $F$ is the number of videos and $L$ is bitrate level. The $X(t^x_y)$ can be further expressed as $X(t^x_y) \in \{0, 1\}_{F \times L}$. Then caching state variable of video $f_i$ with bitrate level $\ell_j$ at the time slot $t^y_x$ is expressed as

$$x_{f_i, \ell_j}(t^y_x) = \begin{cases} 1, & \text{if MECS cached the } \ell_j \text{ level of video } f_i \\ 0, & \text{otherwise} \end{cases}. \quad (8)$$

The video caching state vector is defined as $\chi_{f}(t^y_x) = \{x_{f_1, \ell_1}(t^y_x), x_{f_2, \ell_2}(t^y_x), \ldots, x_{f_F, \ell_F}(t^y_x)\}$, where $\chi_{f_i}(t^y_x)$ is the number of video copy of $f_i$ in MECS cached at the time slot $t^y_x$.

$$\chi_{f_i}(t^y_x) = \sum_{j \in L} x_{f_i, \ell_j}(t^y_x) \leq 1. \quad (9)$$

The Formula (9) is the constraint that the MECS can only cache one video copy of the bitrate for the same video simultaneously. Additionally, existing research shows that low bitrate video copies can be obtained from high bitrate video transcoding by MECS. Hence, multiple video copies which are cached simultaneously will cause heavy waste of caching resources [34]. Therefore, the constraint condition of the caching capacity of MECS is expressed as

$$\sum_{i \in F} \sum_{j \in L} g_{ij} x_{f_i, \ell_j}(t^y_x) \leq \varphi_{\text{MEC}}. \quad (10)$$

Based on Zipf’s law, the video’s requested probability of $F$ can be denoted as $P^F_x(f) = \{P^F_x(f_1), P^F_x(f_2), \ldots, P^F_x(f_F)\}$. Then video $f_i$ is the requested probability by the vehicle at the time slot $t^x$ and can be expressed as

$$P^F_x(f_i) = \left( f_i^{\mu(t^x)} \sum_{k=1}^{F} f_k^{-\mu(t^x)} \right)^{-1}, \quad (11)$$

where $\mu(t^x)$ is popularity parameter at the time slot $t^x$, which reflects the shape of video popularity distribution. Without loss of generality, suppose $\mu(t^x)$ obeys a Markov process $\mu(t^x) \in \{\mu_1, \mu_2, \ldots, \mu_G\}$, which containing $G$ parameter. Let $\phi_{i,j}$ denote transition probability from popularity parameter $\mu_i$ at the time slot $t^x$ to $\mu_j$ at time $t^{x+1}$. Then the state transition probability matrix $\Phi$ of the video popularity parameter $\mu$ is calculated as

$$\Phi = \begin{pmatrix} \phi_{1,1} & \cdots & \phi_{1,G} \\ \vdots & \ddots & \vdots \\ \phi_{G,1} & \cdots & \phi_{G,G} \end{pmatrix}. \quad (12)$$

Hence, the cumulative distribution function of the top $S$ videos in $F$ requested by the VUs is expressed as

$$P^F_{\text{CDF}}(S) = \sum_{i=1}^{S} P^F_x(f_i) = \sum_{i=1}^{S} \frac{f_i^{\mu(t^x)}}{\sum_{k=1}^{F} f_k^{-\mu(t^x)}}. \quad (13)$$

When the caching state matrix is $X(t^y_x)$, and caching vector is $\chi_{f}(t^y_x)$, then the caching hit rate at the time slot $t^y_x$ is expressed as

$$H(t^y_x) = \sum_{i=1}^{F} P^F_x(f_i) \chi_{f_i}(t^y_x). \quad (14)$$

Hence, the $H(t^y_x)$ can be rewritten as
\[
\mathcal{H}(t_g^y) = \sum_{i=1}^{F} P_{f}(f_i) \mathcal{X}_{f_i}(t_g^y) = \frac{\sum_{i=1}^{F} f_i^{-\mu(t_i)} \mathcal{X}_{f_i}(t_g^y)}{\sum_{k=1}^{F} f_k^{-\mu(t_k)}}.
\] (15)

Then the caching loss rate at the time slot \( t_g^y \) is calculated as

\[
\mathcal{H}(t_g^y) = 1 - \mathcal{H}(t_g^y) = \frac{\sum_{i=1}^{F} f_i^{-\mu(t_i)} (1 - \mathcal{X}_{f_i}(t_g^y))}{\sum_{k=1}^{F} f_k^{-\mu(t_k)}}.
\] (16)

3.2. Communication Model

We assume that the access system of the vehicular network is based on orthogonal frequency division multiple access (OFDMA). Therefore, every vehicle linked to RSU or MBS is assigned an orthogonal subchannel. Thus, we do not consider interference among different links. For simplicity, the channel gains are assumed to remain constant during one video delivery period. They have the same distribution, so it is sufficient to concentrate on one vehicle user to study the performance of interest. The transmission rate between one vehicle user and RSU at time slot \( t_g^y \) can be calculated by

\[
\mathcal{R}_{MEC}(t_g^y) = W_r \left( \log_2 \left( 1 + \frac{P_r |h_r(t_g^y)|^2}{\delta^2} \right) \right),
\] (17)

where \( W_r \) is the channel bandwidth allocated by RSU to vehicle users, \( P_r \) is the signal transmit power of RSU, \( \delta^2 \) is the noise power, \( h_r(t_g^y) \) is the RSU channel gain at time slot \( t_g^y \), it can be expressed as

\[
|h_r(t_g^y)|^2 = G_r d_{1,u}(t_g^y)^{-\epsilon} |h_0(t_g^y)|^2,
\] (18)

where \( \epsilon \) is the path loss coefficient, \( h_0(t_g^y) \) is a complex Gaussian distributed random variable for Rayleigh channel fading, \( |\cdot| \) is absolute operation, \( |h_0(t_g^y)|^2 \sim \text{CN}(0,1) \), \( G_r \) is the antenna gain coefficient of RSU, \( d_{1,u}(t_g^y) \) represents the distance between the vehicle user and RSU. If the MECs does not cache any bitrate video copies of the video requested by the vehicle user, MBS will take over the service of the vehicle user through the backhaul network, and distribute video copies with the highest bitrate under the constrained backhaul bandwidth to the VUs. The transmission rate between the vehicle user and MBS is expressed as

\[
\mathcal{R}_{MBS}(t_g^y) = B_0 \left( \log_2 \left( 1 + \frac{P_{bs} |h_{bs}(t_g^y)|^2}{\delta^2} \right) \right),
\] (19)

where \( B_0 \) is MBS allocates bandwidth to each vehicle user, \( P_{bs} \) is the MBS transmit power, MBS channel gain at the time slot \( t_g^y \) is calculated as

\[
|h_{bs}(t_g^y)|^2 = G_{bs} d_{0,u}(t_g^y)^{-\epsilon} |h_0(t_g^y)|^2
\] (20)

where \( G_{bs} \) is the antenna gain coefficient of MBS, \( d_{0,u}(t_g^y) \) represents the distance between the vehicle user and MBS. Due to \( \mathcal{R}_{MBS} \) is both limited by the wireless access network and the backhaul bandwidth \( C_{BM} \), when the number of vehicle users \( N(t_g^y) \) connected to
MBS at time \( t_{y,z}^x \) does not reach the maximum number of bearer users \( N_{\text{Max}} \) of MBS, the principle of fairness will be taken into consideration. The bandwidth resource required for caching update will allocate a size of \( C_{\text{MBS}} \) backhaul bandwidth to each vehicle user. \( N_{\text{Max}} \) is defined as

\[
N_{\text{Max}} = \left\lfloor \frac{C_{\text{BM}}}{C_{\text{MBS}}} \right\rfloor, \tag{21}
\]

where \( \lfloor . \rfloor \) is the rounding down operation. When the number of vehicle users connected to MBS exceeds \( N_{\text{Max}} \), the backhaul resources will be average allocated to users, and the transmission rate \( R_{\text{Back}}(t_{y,z}^x) \) can be expressed as

\[
R_{\text{Back}}(t_{y,z}^x) = \begin{cases} 
C_{\text{MBS}}, & N(t_{y,z}^x) \leq N_{\text{Max}} \\
\frac{C_{\text{BM}}}{N(t_{y,z}^x)}, & N(t_{y,z}^x) > N_{\text{Max}}.
\end{cases}
\tag{22}
\]

Hence, the service state probability can be expressed as

\[
P_{\text{AR}}(U(t_{y,z}^x), N(t_{y,z}^x)) = \frac{U(t_{y,z}^x)}{N(t_{y,z}^x)} \cdot \mathcal{H}(t_{y}^x) \left[ U(t_{y,z}^x) - N(t_{y,z}^x) \right]^+ \cdot \mathcal{H}(t_{y}^x) N(t_{y,z}^x). \tag{23}
\]

where \( U(t_{y,z}^x) \) is the number of vehicle users at the time slot \( t_{y,z}^x \), \( N(t_{y,z}^x) \) is the number of vehicle users who download the video by the backhaul link. The average transmission rate by backhaul link at the time \( t_{y,z}^x \) is calculated as

\[
R_{\text{BH}}(t_{y,z}^x) = \frac{\mathbb{E}[U(t_{y,z}^x) - N(t_{y,z}^x)]}{\mathbb{E}[U(t_{y,z}^x)] - N(t_{y,z}^x)} R_{\text{Back}}(t_{y,z}^x). \tag{24}
\]

where \( \mathbb{E} \) is the expect operation. Then, the transmission rate of MBS can be rewritten as

\[
R_{\text{BH}}(t_{y,z}^x) = \min \{ R_{\text{MBS}}(t_{y,z}^x), R_{\text{Back}}(t_{y,z}^x) \}. \tag{25}
\]

### 3.3. ADP Weighted MV-MOS Model

In summary, the multiple bitrate video distribution strategy is that if the video requested by the vehicle user has been pre-cached in the MECS, the video file with the highest bitrate within the MECS will be sent to the vehicle user directly. Otherwise, VUs request the lowest bitrate replicas of videos via the MBS. The intelligent caching in IoV problem aims to maximize vehicle users’ QoE. The MV-MOS is a mobile video experience metric at the device level, it has been widely used in the QoE evaluation of mobile video, e.g., \([33, 34, 38]\)), so it is expressed as

\[
f_{\text{MOS}}(\ell_j, R_{\ell_j}) = M_V(\ell_j, R_{\ell_j}). \tag{26}
\]

where \( M_V(\ell_j, R_{\ell_j}) \) is required data transmission rate, \( \ell_j \) is the video definition level, and \( R_{\ell_j} \) is the minimum bitrate requirement corresponding to the video definition level \( \ell_j \). Then mobile video mean opinion score for the vehicle user to obtain the video from the MECS at the time \( t_{y,z}^x \) is calculated as

\[
Q_{\text{MEC}}(t_{y,z}^x) = \sum_{i \in F} \sum_{j \in L} x_{i,j,\ell_j} \left( t_{y,z}^x \right) P_{f}^x(f_i) f_{\text{MOS}}(\ell_j, \mathcal{R}_{\text{MEC}}(t_{y,z}^x)). \tag{27}
\]
Hence, mobile video mean opinion score for the vehicle user to obtain a video from MBS at time slot $t^x_{y,z}$ can be expressed as

$$Q_{BH}(t^x_{y,z}) = \sum_{i \in F} \left( 1 - \chi_{f_i}(t^x_{y,z}) \right) P^f_{x,y}(f_i) \max f_{MOS} \left( \ell_f, R_{BH}(t^x_{y,z}) \right). \quad (28)$$

When the number of users in the service coverage area at time slot $t^x_{y,z}$ is $U(t^x_{y,z})$, and the number of VUs downloading videos through MBS is $N(t^x_{y,z})$, all are obtained from the MECS, mobile video mean opinion score for the vehicle user is expressed as

$$Q_M \left( U(t^x_{y,z}), N(t^x_{y,z}) \right) = \left( U(t^x_{y,z}) - N(t^x_{y,z}) \right) P_{Ar} \left( U(t^x_{y,z}), N(t^x_{y,z}) \right) Q_{MEC}(t^x_{y,z}). \quad (29)$$

The mobile video mean opinion score for the $N(t^x_{y,z})$ VUs who obtain videos from MBS through the backhaul network at the time slot $t^x_{y,z}$ is expressed as

$$Q_B \left( U(t^x_{y,z}), N(t^x_{y,z}) \right) = P_{Ar} \left( U(t^x_{y,z}), N(t^x_{y,z}) \right) N(t^x_{y,z}) Q_{BH}(t^x_{y,z}). \quad (30)$$

When the number of vehicle users via the RSU service at the time slot $t^x_{y,z}$ is $U(t^x_{y,z})$, and the number of vehicle users downloading videos through MBS is $N(t^x_{y,z})$, then the average mobile video mean opinion score at the time slot $t^x_{y,z}$ is expressed as

$$\overline{Q}_{AV} \left( U(t^x_{y,z}), N(t^x_{y,z}) \right) = \frac{Q_M \left( U(t^x_{y,z}), N(t^x_{y,z}) \right) + Q_B \left( U(t^x_{y,z}), N(t^x_{y,z}) \right)}{U(t^x_{y,z})}. \quad (31)$$

Hence, the average mobile video mean opinion score at time slot $t^x_{y,z}$ is expressed as

$$Q_{Aver}(t^x_{y,z}) = \frac{\mathbb{E}[U(t^x_{y,z})]}{N(t^x_{y,z})} \overline{Q}_{AV} \left( U(t^x_{y,z}), N(t^x_{y,z}) \right). \quad (32)$$

Therefore, the average MV-MOS of vehicle users for one period $\Delta t^x_{y,z}$ is expressed as

$$Q_{MOS}(t^x_{y,z}) = \frac{\sum_{z \in K_{SN}} Q_{Aver}(t^x_{y,z})}{K_{SN}}. \quad (33)$$

The multiple bitrate level of mobile video not only affects the resolution of the video, but is also related to the size of the video file. Owing to the service time of RSU for vehicle users is limited, it may cause the vehicle user to only complete a small part of the download task within the time of the RSU service range, which means bad QoE. When the vehicle user leaves the RSU service coverage area, MBS will continue to distribute the unfinished part of the replicas of video. This is very likely to be difficult to meet the QoE requirements of the corresponding bitrate replicas of video due to the low MBS transmission rate, resulting in serious video service fail. To deal with the above problems, it introduces the evaluation index of the ADP of vehicle users into the QoE evaluation model, then proposes an ADP weighted MV-MOS model that comprehensively considers the completion of the video download task. The calculation formula of ADP for mobile video is expressed as
where $\mathcal{R}_a(t^*_{xy})$ is transmission rate, $g_j$ is file size of the video bitrate level $\ell_j$. When vehicle users obtain video from the MECS, the $P_{MEC}(t^*_{xy})$ can be expressed as

$$P_{MEC}(t^*_{xy}) = \sum_{i \in F} \sum_{j \in L} x_{f_i \ell_j}(t^*_{xy}) P_F^i(f_i) P_{ADP}(f_i, \ell_j, \mathcal{R}_{MEC}(t^*_{xy})).$$

When a vehicle user obtains a video from MBS through the backhaul at the time slot $t^*_{xy}$, the $P_{BH}(t^*_{xy})$ is calculated as

$$P_{BH}(t^*_{xy}) = \sum_{i \in F} \sum_{j \in L} x_{f_i \ell_j}(t^*_{xy}) P_F^i(f_i) P_{ADP}(f_i, \ell_j, \mathcal{R}_{BH}(t^*_{xy})).$$

Hence, the average ADP at time slot $t^*_{xy}$ is calculated as

$$\overline{P}_{ADP}(t^*_{xy}) = \sum_{k=0}^{E} P_{AR}(E(\mathcal{U}(t^*_{xy})), K) \left( \frac{KP_{BH}(t^*_{xy})}{E(\mathcal{U}(t^*_{xy}))} + \left( \frac{E(\mathcal{U}(t^*_{xy}))}{E(\mathcal{U}(t^*_{xy}))} - K \right) P_{MEC}(t^*_{xy}) \right).$$

In summary, the ADP weighted MV-MOS at the time slot $t^*_{xy}$ is expressed as

$$J(t^*_{xy}) = \overline{P}_{ADP}(t^*_{xy}) Q_{MOS}(t^*_{xy}).$$

4. Problem Formulation

4.1. ADP Weighted Mobile Video Mean Opinion Score

This section, we focus on the problem of intelligent caching for mobile video streaming in the IoV. In mobile networks, the mobile user’s location is considered static, and then the QoE evaluation method cannot be applied to the dynamic scene of the vehicular network. To adapt to the vehicle’s fast mobility scenario, we formulate the problem of mobile video caching in IoV as an optimized expression under multiple constraints to maximize the ADP weighted MV-MOS. The objective function can be formulated as

$$\mathcal{P} : \arg \max J(t^*_{xy}),$$

s.t. $\sum_{j \in L} x_{f_i \ell_j}(t^*_{xy}) \leq 1,$

$$\sum_{i \in F} \sum_{j \in L} g_j x_{f_i \ell_j}(t^*_{xy}) \leq \varphi_{MEC},$$

$$\mathcal{R}_{BH}(t^*_{xy}) \leq \mathcal{R}_{MEC}(t^*_{xy}),$$

$$\mathcal{R}_{BH}(t^*_{xy}) = \min \{ \mathcal{R}_{BH}(t^*_{xy}), \mathcal{R}_{MBS}(t^*_{xy}) \}.$$
4.2. Average Mobile Video Mean Opinion Score

For better experimental comparison, this section formulates a sub-problem \( P_1 \) for the optimization problem of Formula (39a), which is to relax (39a) and strip the ADP weighting term to replace the optimization objective is Formula (33), so the multi-dimensional constraint expression can be formulated as

\[
P_1: \arg \max_{Q_{MOS}} \left( t_y^x \right),
\]

subject to

\[
s.t. \sum_{f \in L} x_{f,i} \left( t_y^x \right) \leq 1,
\]

\[
\sum_{i \in F} \sum_{j \in L} x_{f,i} \left( t_y^x \right) \leq \varphi_{MEC},
\]

\[
R_{Back}(t_{y,z}^x) \leq R_{MEC}(t_{y,z}^x),
\]

\[
R_{BH}(t_{y,z}^x) = \min \left\{ R_{Back}(t_{y,z}^x), R_{MBS}(t_{y,z}^x) \right\}.
\]

In fact, \( P_1 \) is a simplification of \( P \), then constraint conditions of \( P_1 \) the same as \( P \), but only the optimization objective is different. So the parameters definition and constraint conditions of (40) are the same as (39).

5. Deep Reinforcement Learning-Based Caching Solution

The video content caching problem in this section is formulated as MDP. A novel caching scheme DQN-based is proposed to achieve maximum ADP weighted MV-MOS, to solve this complex problem. Furthermore, we also improve the convergence speed of DQN and enhance the performance of proposed algorithm. The MDP can be represented by the 4-tuple \(<S, A, E, R>\). \( S \) is the set of environment states, \( A \) denotes the set of agent actions, \( E \) represents the state transition probability, and \( R \) indicates the reward function.

5.1. Markov Decision Process

At the beginning, the agent will observe environment information. The environment state is represented as

\[
s_{t_y^x} = \{ \lambda \left( t_y^x \right), \mu \left( t_y^x \right), X \left( t_{y-1}^x \right) \},
\]

The action \( a_{t_y^x} \) is represented as

\[
a_{t_y^x} = \{ X \left( t_y^x \right) \},
\]

After an action \( a_{t_y^x} \) is taken, the reward \( R \left( s_{t_y^x}, a_{t_y^x} \right) \) is represented as

\[
R \left( s_{t_y^x}, a_{t_y^x} \right) = \arg \max E \left\{ \sum_{y=0}^{\infty} \omega^y \mathbb{E} \left\{ s_{t_y^x}, a_{t_y^x} \right\} \right\},
\]

where \( \omega^y \in (0, 1] \) is discounted factor. Hence, the optimal caching strategy \( \pi^* \) in the IoV is expressed as

\[
\pi^* \left( s_{t_y^x} \right) = \arg \max Q \left( s_{t_y^x}, a_{t_y^x} \right).
\]

In summary, the agent perceives the environment state \( s_{t_y^x} \), then selects and executes an action \( a_{t_y^x} \), after the system environment will feedback an immediate reward \( R \left( s_{t_y^x}, a_{t_y^x} \right) \), it is represented as

\[
R \left( s_{t_y^x}, a_{t_y^x} \right) = J \left( t_y^x \right).
\]
Otherwise, the system environment will feedback on a cost (e.g., −1).

5.2. DRL-Based Caching Algorithm

The high mobility of vehicle users leads to environmental information dynamic changes in vehicular networks. To cope with the challenge, we consider a novel algorithm that effectively solves excessive state and action space in a dynamic environment by combining deep learning and reinforcement learning to solve the complex caching problem. Especially, as the update rule of the DQN does not require knowledge about the transition and reward functions. Therefore, we proposed the P3DVQC to solve the caching problem. The parameters of the P3DVQC main networks update formula is expressed as

\[
\theta_{t+1}^{\pi} = \alpha \left( \gamma Q \left( s_{t+1}^{\pi}, \max_{a_{t+1}^{\pi}} Q \left( s_{t+1}^{\pi}, a_{t+1}^{\pi} ; \theta_{t}^{\pi} \right) ; \theta_{t}^{-} \right) + \theta_{t}^{-} - Q \left( s_{t}^{\pi}, a_{t}^{\pi} ; \theta_{t}^{\pi} \right) \right) \nabla Q \left( s_{t}^{\pi}, a_{t}^{\pi} ; \theta_{t}^{\pi} \right) + R(t_{t}^{\pi}),
\]

(46)

where \( \alpha \) is learning rate, \( \gamma \) is discount factor, \( \nabla \) is gradient operator. The P3DVQC is integrated with prioritized experience replay (PER) technology to achieve priority sampling by changing sampling distribution to improve the performance of the DQN. The P3DVQC sampling weight formula is expressed as

\[
p_{i}^{PER} = \frac{\rho_{i}^{PER}}{\sum_{i \in \Omega} \rho_{i}^{PER}},
\]

(47)

where \( \rho_{i}^{PER} \) is priority parameter of experience pool sampling \( i \), it can be expressed as

\[
\rho_{i}^{PER} = \left( \Delta E_{TD}(t_{y}^{\pi}+1) + \varepsilon \right)^{\beta},
\]

(48)

where \( \varepsilon \) is disturbance coefficient, \( \Delta E_{TD}(t_{y}^{\pi}+1) \) is a distance of \( \langle s_{t}^{\pi}, a_{t}^{\pi}, R(t_{t}^{\pi}), s_{t+1}^{\pi} \rangle \) between output and target value. It is calculated as

\[
\Delta E_{TD}(t_{y}^{\pi}+1) = \left| \gamma Q \left( s_{t}^{\pi}, \max_{a_{t}^{\pi}} Q \left( s_{t}^{\pi}, a_{t}^{\pi} ; \theta_{t}^{\pi} \right) ; \theta_{t}^{-} \right) + R(t_{t}^{\pi}) - Q \left( s_{t}^{\pi}, a_{t}^{\pi} ; \theta_{t}^{\pi} \right) \right|,
\]

(49)

where \( \theta_{t}^{\pi} \) is the parameter of P3DVQC main network, \( \theta_{t}^{-} \) is parameter of P3DVQC target network, \( |.| \) is absolute operation. In addition, P3DVQC also integrated dueling technology to improve the deep Q-learning algorithm. The framework of the proposed P3DVQC is shown in Figure 3. The target network output function is expression as

\[
Q \left( s_{t}^{\pi}, a_{t}^{\pi} ; \theta_{t}^{\pi}, a_{t}^{\pi}, \beta_{t}^{\pi} \right) = V \left( s_{t}^{\pi} ; \beta_{t}^{\pi} \right) + A \left( s_{t}^{\pi}, a_{t}^{\pi} ; \theta_{t}^{\pi}, a_{t}^{\pi} \right),
\]

(50)

where \( a_{t}^{\pi} \) is parameter of advantage function \( A \left( s_{t}^{\pi}, a_{t}^{\pi} ; \theta_{t}^{\pi}, a_{t}^{\pi} \right) \), \( \beta_{t}^{\pi} \) is parameter of value function \( V \left( s_{t}^{\pi} ; \beta_{t}^{\pi} \right) \). The pseudo code of the proposed DRL-based caching algorithm is provided in Algorithm 1.
Algorithm 1 Prioritized experience replay Dueling Double DQN IoV QoE Caching (P3DVQC)

1. **Initialize:**
   1. \( s_{t} \in S \), \( a_{t} \in A \), \( \alpha \), \( \gamma \), the size of minibatch \( G \) and the memory pool \( N \)
   2. Parameters \( \theta_{xy} \) and \( \theta_{-xy} \)
   3. Parameters \( \Delta E_{TD}(t_{xy}^{+}) \)
   4. Parameters \( V(s_{t}; \theta_{xy}, \beta_{xy}), A(s_{t}, a_{t}; \theta_{xy}, \alpha_{xy}) \)

2. **Learning:**
   7. for \( t_{xy}^{+} \in K \) do
   8. Choose the policy \( \pi(t_{xy}^{+}) \) by \( \epsilon - greedy \) policy
   9. Observe environment, evaluate and estimate \( s_{t} \)
   10. Perform \( a_{t} \) according to \( \pi \) and observe feedback
   11. while \( t_{xy}, z \in T_{s} \) do
   12. Observe: environment and communication state
   13. Calculate: \( H(t_{y}^{+}), h_{10}(t_{y}^{+}), R_{MEC}(t_{y}^{+}), R_{MBS}(t_{y}^{+}) \)
   14. Obtain: \( Q_{AV}(U(t_{y}^{+}, N(t_{y}^{+})), Q_{Aver}(t_{y}^{+}) \)
   15. end while
   16. Calculate: \( R(s_{t}^{+}, a_{t}^{+}, R(t_{y}^{+}), s_{t+1}^{+}) \)
   17. for \( j \in N_{F} \) do
   18. Calculate: \( \rho_{PER}^{+} \) sort by SumTree, and sample \( <s_{t}^{+}, a_{t}^{+}, R(t_{y}^{+}), s_{t+1}^{+}> \)
   19. Update: \( \Delta E_{TD}(t_{xy}^{+}) \) and function parameters
   20. Update: \( \theta_{xy}^{+} \) with SGD algorithm
   21. Update: \( Q, V, A, \theta_{xy}^{+} \), and network parameters
   22. end for
   23. end if
   24. end for

Figure 3. The proposed framework.
6. Numerical Results and Discussion

The simulations of the caching scheme are carried out in this section. The performance of P3DVQC was compared with the baseline schemes. The simulation environment for the mobile video caching system in IoV was programmed in Python. In addition, the TensorFlow platform was used to implement the P3DVQC caching scheme based on the open-source package convolutional neural networks. The main system parameters used in the simulations are summarized in Table 1. For performance comparison, five benchmark schemes were presented:

1. Resolution Optimal Caching Scheme (ROCS): The scheme always priority caches the video copies with the highest bitrate level to realize the rapid delivery of high-quality video content and improve the quality of experience of vehicle users.
2. Fluency Optimal Caching Scheme (FOCS): The scheme caches the video copies with the lowest bitrate to cache as many videos as possible and realizes the diversity of the cache to improve the cache hit rate.
3. Random Caching Scheme (RCS): The scheme selects the video copies with the random algorithm, which means an equal probability choose each copy cached until the maximum cache capacity of MECS is reached.
4. Cost Efficient Scheme (CES): The scheme only depends on the backhaul network delivery of video files, not to use of MECS caching equipment to minimize device cost and energy consumption.
5. Brute Force Scheme (BFS): The scheme is optimal, but it is obtained based on ideal conditions where all system information is known. In actual scenarios, system information, such as wireless channel state information, cannot be obtained in advance. Therefore, BFS cannot carry out in real vehicular network scenarios.

Table 1. System Parameters.

<table>
<thead>
<tr>
<th>Notation</th>
<th>Definition</th>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P_{bs}$</td>
<td>40 dbm</td>
<td>$\varphi_{MEC}$</td>
<td>[500, 2000] Mb</td>
</tr>
<tr>
<td>$\gamma$</td>
<td>[0.1, 0.9]</td>
<td>$\beta$</td>
<td>[0.5, 0.8]</td>
</tr>
<tr>
<td>$\mu_m$</td>
<td>[0.4, 1.5]</td>
<td>$\gamma$</td>
<td>(0, 1)</td>
</tr>
<tr>
<td>$\epsilon$</td>
<td>4</td>
<td>$\delta^2$</td>
<td>$-102$ dBm</td>
</tr>
<tr>
<td>$P_r$</td>
<td>33 dbm</td>
<td>$L$</td>
<td>[5, 10]</td>
</tr>
<tr>
<td>$\theta$</td>
<td>$\pi/6$</td>
<td>$C_{MBS}$</td>
<td>[3, 6] Mbps</td>
</tr>
<tr>
<td>$\mathcal{F}$</td>
<td>[10, 50]</td>
<td>$C_{BM}$</td>
<td>[10, 100] Mbps</td>
</tr>
</tbody>
</table>

6.1. Algorithm Analysis

Figure 4 shows the performance of the two DRL-based caching algorithms proposed. One is based on the traditional deep Q-network IoV caching algorithm (DVQC), and another is the P3DVQC algorithm which integrates the prioritized experience replay, the dueling, and the double deep Q-network technologies. The size of the experience replay pool is 3000 in the simulation. Figure 4 also shows that the P3DVQC completes the convergence at nearly 5000-time slots, while the DVQC at about 6000-time slots. That is due to the P3DVQC employing the PER technology improving the deep neural network training efficiency. In addition, the P3DVQC shows more stability than DVQC at the convergence state, this is because the P3DVQC integrated DDQN and Dueling technology, over-estimation of DQN is effectively eliminated, avoiding unnecessary misselections to achieve rapid convergence to the optimal caching strategy.

Figure 5 shows the convergence performance comparison between the P3DVQC and the DVQC. Before 3000 time slots is the experience replay pool initialization stage. After 3000 time slots, the P3DVQC achieves a faster convergence rate than DVQC. That is due to the P3DVQC employing the PER to achieve TD weighting of the experience pool samples, which makes deep neural networks more effective. The P3DVQC algorithm determines the priority of the sample selected for training according to the size of TD errors to improve the training effect and speed up the convergence speed. However, the DVQC algorithm
selects samples for training through a uniform sampling strategy, so the convergence speed is slower than P3DVQC.

Figure 4. A comparison of P3DVQC and DVQC.

Figure 5. Convergence speed.

6.2. Average MV-MOS Performance

Figure 6 shows the average MV-MOS performance comparison between the P3DVQC scheme and the five baseline schemes. BFS is the ideal upper bound algorithm of the experiment, it is obtained by traversing all the solution spaces and requires knowing all the system information, which requires a lot of time overhead to search the solution space, hardly realized in the actual scene. The CES scheme does not use any caching technology, so its performance is the lowest. Secondly, the RCS chooses video content caching by the random algorithm, resulting in the performance being very unstable. The FOCS only caches the lowest bit rate video. Because the lowest bit rate video file is small, the FOCS can cache more different video content varieties, but it cannot effectively provide high-definition video. ROCS caches high-bitrate HD videos as much as possible, so vehicle users can achieve high average MV-MOS if the required HD video is pre-cached in MECS. However, the size of high bitrate video content is larger, so the higher storage overhead leads to a lower cache hit rate and performance. The P3DVQC proposed by this paper comprehensively considers the multiple bitrate video quality, backhaul bandwidth, and caching capacity to maximize objective. Therefore, the caching strategy obtained by the P3DVQC is that it caches a mixture of low and high bitrates. Figure 6 shows that the proposed can converge to the upper bound, and its performance is better than other benchmark algorithms.
6.3. ADP Weighted MV-MOS Performance

Figure 7 shows that ADP weighted MV-MOS performance indicators by different caching schemes. The ROCS scheme achieves a poor performance in ADP weighted MV-MOS, almost the same as the CES algorithm. This is because the video’s bitrate is directly proportional to the file size. A high bitrate means a large storage cost. Hence, when the service time is fixed, the ROCS can only complete a small proportion of tasks, which will cause vehicle users to only enjoy a short time of the high-definition video within the service area. It will immediately interrupt when it leaves the caching service area. Therefore, based on the average MV-MOS model cannot be directly applied to the vehicular networks. Figure 7 also shows that the ADP weighted MV-MOS performance of the P3DVQC can converges to the optimal BFS, which is better than other benchmark algorithms.

Figure 8 shows the cumulative reward of the five algorithms in Figure 7. The BFS is a straight line in Figure 7, so the cumulative gain is meaningless. To increase the caching hit rate, FOCS only caches the lowest bitrate videos so that its solution space is much smaller than the P3DVQC, so the cumulative return obtained in the early stage is more significant. However, with the P3DVQC converging to the optimal strategy, it achieves a greater cumulative return than the FOCS. Other benchmark algorithms can be seen from the analysis of Figure 7, and their performance is lower than the P3DVQC.
7. Conclusions

In this paper, a DRL-based P3DVQC algorithm is proposed to solve the mobile video caching problem while achieving the maximum ADP weighted MV-MOS. The numerical results show that compared with other benchmark schemes, the proposed has a faster convergence speed and significant performance.

Author Contributions: Z.L.: investigation, methodology, and writing—original draft. M.L.: writing—review & editing. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the Shenzhen and Hong Kong Joint Program of Shenzhen Scientific Plan under Grant SCDX2020103095406023, in part by the Basic Research Program of Shenzhen Scientific Plan under Grant JCYJ20180507182446643, in part by the National Natural Science Foundation of China under Grant 62271424, and in part by the National Natural Science Foundation of China under Grant 61871339.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

Abbreviations

The following abbreviations are used in this manuscript:

- 5G: 5th Generation
- MEC: Multi-access Edge Computing
- DRL: Deep Reinforcement Learning
- QoE: Quality of Experience
- QoS: Quality of Service
- MOS: Mean Opinion Score
- MV-MOS: Mobile Video Mean Opinion Score
- ADP: Average Download Percentage
- RL: Reinforcement Learning
- RSU: Road Side Unit
- MECS: MEC server
- IoV: Internet of Vehicles
- MBS: Macro Base Station
- MDP: Markov decision process
- C-V2X: Cellular Vehicle-to-everything
- DQN: Deep Q-learning Network
- SBS: Small Base Stations
<table>
<thead>
<tr>
<th>CBR</th>
<th>Constant Bitrate</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVC</td>
<td>Scalable Video Coding</td>
</tr>
<tr>
<td>ABR</td>
<td>Adaptive Bitrate</td>
</tr>
<tr>
<td>VUs</td>
<td>Vehicle users</td>
</tr>
<tr>
<td>HD</td>
<td>High-definition</td>
</tr>
</tbody>
</table>

**References**


