Journal of Artificial Intelligence and Data Mining (JAIDM), Vol. 10, No. 3, 2022, 411-422.

Shahrood University of Technology

Journal of Artificial Intelligence and Data Mining (JAIDM)
Journal homepage: http://jad.shahroodut.ac.ir

Research paper

Q-LVS: A Q-Learning-based Algorithm for Video Streaming in Peer-to-Peer Networks Considering a Token-Based Incentive Mechanism

Zahra ImaniMehr*

Computer Engineering, Amir Kabir University of Technology, Tehran, Iran.

Article Info

Article History:
Received 12 October 2021
Revised 19 February 2022
Accepted 06 April 2022

DOI:10.22044/jadm.2022.11293.2287

Keywords:
Peer-to-peer Networks, Layered Video Coding, Token, Incentive, Q-learning, Continuous Markov Decision process.

**Corresponding author: Zimanimehr@aut.ac.ir (Z. ImaniMehr).*

Abstract

The peer-to-peer video streaming has reached great attention during the recent years. Video streaming in the peer-to-peer networks is a good way to stream video on the Internet due to the high scalability, high video quality, and low bandwidth requirements. In this paper, the issue of live video streaming in the peer-to-peer networks that contain selfish peers is addressed. In order to encourage the peers to cooperate in video distribution, tokens are used as an internal currency. The tokens are gained by the peers when they accept requests from other peers to upload video chunks to them, and tokens are spent when sending requests to other peers to download video chunks from them. In order to handle the heterogeneity in the bandwidth of peers, the assumption has been made that the video is coded as multi-layered. For each layer, the same token has been used but priced differently per layer. Based on the available token pools, the peers can request various qualities. A new token-based incentive mechanism has been proposed that adapts the admission control policy of the peers according to the dynamics of the request submission, request arrival, time to send requests, and bandwidth availability processes. The peer-to-peer requests could arrive at any time, so the continuous Markov decision process has been employed.

1. Introduction

One of the most important and challenging applications in the peer-to-peer networks [1] is video streaming. This application requires a maximum participation of the peers. If the participation of peers in this application is low, the video quality is greatly reduced. In a peer-to-peer network, nodes exchange video chunks. Unlike downloading, uploading is costly for the peers since they are required to use their own upload bandwidth. On the other hand, the main premise of these networks is altruistic resource sharing.

The peer-to-peer networks are based on the participation of peers. However, selfish peers like to use the network's resources but do not share anything due to the cost attached to sharing resources. In fact, a selfish node in the network is a free-rider that attends only to its own benefit regardless of the overall system performance [2]. In the Napster and Gnutella network, 25% of peers on the network share no files at all [3].

Accordingly, there is a need for a mechanism to incentivize the peers to participate in video sharing. Efforts to solve the problem of free riding and incentivize cooperation in selfish peers have led to the introduction of three different classes of methods [4]: 1) barter-based mechanism [5-8], 2) reputation-based mechanism [9-11], 3) payment-based mechanism [12-16].

The peer-to-peer networks have two types of structures: tree-based and mesh-based. In a tree-based structure, the peers fulfil the roles of parent and child. Maintenance of this structure is costly and due to the continuous arrival and departure of nodes in the peer-to-peer networks, the use of this structure is not optimal. In a mesh-based structure,

every peer has several neighbours to exchange data with, and this increases the robustness of the network, and is more popular in the peer-to-peer networks.

In this paper, a token-based method, which falls into the payment-based category, is proposed to incentivize peers in a video streaming application. This method is based on a mesh infrastructure peer-to-peer network. Since the peers have a variety of upload and download bandwidths, layered video coding has been used. Finally, based on their available resources, the peers can request a different number of layers to gain their desired video quality.

The proposed method adapts the admission control policy of the peers according to the dynamics of four parameters: the request submission, request arrival, time to send requests, and bandwidth availability processes. The assumed model for arriving requests fits the continuous Markov Decision Process (MDP), which implies that the peer-to-peer requests can arrive at any time, which is closest to the reality.

When the network is large and the number of peers is high, then the probability of peers interacting with each other in the future decreases, and as a result, interactions between the peers become asymmetric. Asymmetric interactions in the peer-to-peer networks create the possibility of asymmetry in interests, and many of these networks allow the peers to constantly change their identities. When interactions are random, in a token-based manner, since payments are made after data exchange, there is no need for the peers to interact again. In this way, as the basis for interaction is payment, there is no need for symmetry in interactions.

Finally, the contributions of this work are briefly listed as follow:

- The token-based method is presented to motivate collaboration between the peers, and the same token is used for all layers to exchange video chunks. The price of each video chunk layer is already known, and the video chunks of the lower layers have a higher price.
- The Q-learning algorithm is used, so that any peer is able to balance the available tokens with bandwidth cost, while still being able to watch high-quality videos.
- The proposed model-free learning algorithm receives real-time feedback from the environment dynamics, and learns the optimal policy. Based on the randomness of the environment, a number of important variables are considered to find the optimal policy including the probability of receiving requests, the probability of sending requests, the amount a peer benefits from watching a video chunk, the cost incurred by sending a video chunk, the size of available bandwidth, and the length of time it takes the peer to send requests.

The rest of this paper is structured as what follows. Section 2 investigates some of the previous research works related to this problem. Section 3 introduces the system model, and presents the optimal policies for the peers, and proposes the Q-LVS algorithm. Section 4 provides the simulation results, and finally, Section 5 concludes the paper by suggesting some future directions.

2. Previous Works

As mentioned earlier, studies related to video streaming in the peer-to-peer networks are mostly categorized into three classes: 1) barter-based mechanisms [5-8], 2) reputation-based mechanisms [9-11], and 3) payment-based mechanisms [12-16].

In a barter-based mechanism, the previous interactions of the peer with other peers form the basis of their current action. Due to the asymmetric nature of interactions in the networks that contain a large amount of peers who exhibit rare and random interactions with other peers, these methods are not applicable. In order to analyse the peer interactions, an infinitely repeating game has been developed in [5] and [6]. The concept of the Nash equilibrium was then used for game analysis. An incentive protocol has been developed in [7] using social norms for multimedia sharing as well as a reputation scheme for use by a peer-to-peer network. Finally, the Chainsaw protocol has been investigated in [8], in which incentives are used to prevent selfish nodes. This incentive scheme is based on the iterated prisoner's dilemma.

In a reputation-based approach, the reputation gained by each peer is based on its level of participation within the network. This reputation also decides the behaviour of the other peers towards this peer. In the indirect reciprocity method, which has a centralized nature, it is required to be determined how to deal with the peers unknown to the other peers ("strangers") while encouraging them to improve the system and preventing the peers from being able to log in and out of the system repeatedly with different

identifiers that would disrupt system operations. However, this method is still vulnerable to Sybil attacks [17], in which a user creates multiple identifiers that are then used to make more profit from the system [17].

In [9], a repeated game-based incentive mechanism has been considered. The game uses a trusted third party to record the participation of peers in each round of the peer-to-peer system. Finally, in [10] a peer-to-peer selection algorithm has been provided for the services that give the peers the option to increase their viewing quality by allowing them more flexibility in selecting their neighbours based on their own level of participation. In [11], a reputation-based approach has been proposed while "Sybil attack detection" helps to facilitate file sharing on a heterogeneous peer-to-peer network. In this way, some restrictions are assigned to Sybil nodes, and free riders are blocked. One of the factors that reputation is based on is the amount of shared bandwidth.

In the payment-based methods [12-15], money is earned by the peers by uploading content to the other peers. Additionally, the downloading content from the other peers has to be paid for. In a study based on virtual currency, an incentive method has been proposed that is modelled as a stochastic game [12]. The premise that this study was based on was that the network was not dynamic. In other words, the neighbours are fixed and known to each peer. However, this is both a limiting as well as an unrealistic assumption for video streaming on a peer-to-peer network.

A dynamic payment-based incentive mechanism for live video streaming has been introduced. In this mechanism, the peers can adjust their income level by adapting it to the process of random incoming request arrivals, their local request processes, and their available bandwidth [13]. However, in [13], the time of requests is not continuous. Additionally, one of the important parameters in finding the optimal policy is the time it takes to respond to the requests, and this was not discussed. In this paper, this has been addressed using a continuous MDP, and furthermore, this parameter has been taken into consideration.

In our prior work [14], a token-based system was used to incentivize the peers to exchange video chunks with each other. This showed that if the expected future benefit of having an additional token outweighs the immediate cost of accepting incoming request, then a self-interested peer would be willing to accept the request. This proved that the threshold strategies were the only strategies that a self-interested peer, who wanted to maximize its own utility, would adopt. However, in [14], this was not addressed that how the individual peers could dynamically optimize their cooperation policies in a non-idealized network environment. A payment-based approach to overcome free riding on the peer-to-peer networks in [15] was introduced. The peers are initially given equal points that increase by one point when a peer uploads and decreases by one point when a peer downloads. If the point balance for a peer reaches a certain amount, the peer is not allowed to download files, and is given a deadline until which the peer can pay its balance. If the balance is not paid, interest is requested from the peer. Additionally, [16] also proposes a payment-based method for preventing free-riding in the peer-to-peer systems. The free-riding peers are punished by losing points if they do not participate in the next round as the providers. The drawback of [15-16] is that although they are payment-based, they have a centralized nature.

In order to conclude this section, the proposed method in this paper is compared with some of the most relevant previous studies. The proposed method falls into the third category, payment-based methods. In this work, several new parameters have been considered including:

- probability of receiving requests
- probability of sending requests
- amount a peer benefits from watching a video chunk
- cost incurred by sending a video chunk
- size of available bandwidth
- length of time it takes the peer to send requests

3. Proposed Method

In the following chapter, the assumptions and the problem formulation are described first. Next, the Q-learning algorithm is introduced, and finally, the proposed method is explained.

3.1 Assumptions and problem formulation

In Figure 1, a sample buffer map is shown. The size of the buffer map is 4, and video has been encoded in four layers. In this figure, each cell shows the status of the video chunk of that layer and time step. Grey cells show the video chunk of that layer received and its time step. The D cells show the video chunks that can be played successfully if they are received. In the proposed method, the peers can send requests for video chunks to its neighbours based on its own buffer maps and its neighbours' buffer maps, which are

shared. For example, the peer in Figure 1 can request a video chunk of layer 3 of time step t+1 from its neighbours if they have this video chunk. Tokens are gained when the peers accept requests from other peers and upload data to them. On the other hand, tokens are given to the peer's neighbours by sending requests and downloading data from them. Data can only be downloaded when the peers have the required amount of tokens. Therefore, data can only be downloaded if the peers have first uploaded data and received tokens.

Both of the base layers and enhancement layers of a video impact the quality of the video in streaming applications. The base layer has a higher impact on the video's quality than the enhancement layers (the higher number of the layer, the lower its impact on the video. The number of the layer has an inverse correlation with the quality of the video). Each individual chunk of a specific video layer is priced differently with the lower layers having a higher price (see Figure 1). In Figure 2, a sample node and its neighbours are shown [14]. In this paper, this assumption has been made that the video has between 1 and 7 layer(s).

With an upload bandwidth of B, any peer can answer a small number of arrival requests. $S0$ is defined as $S0 = \{s(1), s(2), .., s(L)\}$, where $s(l)$ for $1 \leq l \leq L$ is equal to the number of requests sent by the peer for video chunks of the layer *l*.

The assumption is made that peer *i* requests a video chunk of layer l, $1 \leq l \leq L$, (p'th chunk of video) from peer *j*. If peer *j* accepts that request, it must pay $_{(p,)}$ cost based on the size of the requested chunk. This is because peer *j* must consume its upload bandwidth while peer *i*, with respect to the effect of the video chunk on the video quality, will be benefited and this benefit equals $_{(p,)}$. Thus for all video chunks, $(p,) > (p,)$.

The problem of determining the optimal policy for each peer is formulated using MDP. MDP is a five-tuple (S, A, P, R, β), which is defined as the following:

- State space, *S*: that is defined as:

$$S = (token\#, S0) \tag{1}$$

in which $token\#$ is equal to the sum of tokens that the peer holds, and $S0$ is a vector that represents the number of sending requests that are related to the different layers. Whenever a peer receives a request, in addition to deciding on the number of existing tokens, it is necessary to check whether there is enough bandwidth to upload or not.

- Actions, *A*: Each peer can perform actions by entering an event. The action space is defined as $A = \{R, A_r, D_1, D_2, \dots, D_L\}$ such that $a = R$ implies rejection of the request and not sending a request if there is any; $a = A_r$ implies accepting an incoming request and not sending a request if it exists, and $a = D_l$ implies requesting the video chunk of layer l from another peer and accepting the request of that peer.
- Probability, P: $P_a(S.S') = \Pr(S_{t+1} = S' | S_t = s. a_t = a)$ is the probability that the action a in the state S in the time interval t leads to the transition to the state S' in the time interval $t + 1$.
- Reward, *R*: The peer will incur c_l cost when accepting the request of layer l and will have the benefit of b_l when sending a request of layer l to another peer and acceptance of it according to:

$$r_t = \begin{cases} -c_l, & if\ a_t = A_l \\ b_l, & if\ a_t = D_l \\ 0, & if\ a_t = R \end{cases} \tag{2}$$

- Discount factor, β: that represents the difference in importance between the future rewards and present rewards.

The peer-to-peer requests could arrive at any time, so the Continuous-time Markov Decision Process (CMDP) modelling is used. When the peers exchange their buffer map, the multiple peers may send their requests to a peer. In order to examine all of them, a queue is required. The problem that arises is what the optimal length of the queue is and how long the request of each peer can wait in the queue. These two parameters play a key role since their values affect the effectiveness of the program. In order to solve this problem and to match the problem to reality, the time is considered to be continuous. It is assumed that the peers exchange their buffer map at specific intervals, and based on that, they send their requests to the peer who has the video chunks they need.

The input requests are assumed to follow the Poisson distribution, and the request entrance rate for each layer is equal to $\alpha(l)$. If the request is

accepted, the average time it takes to send the request to the requesting peer is equal to $1/\gamma(l)$.

The input requests of each layer follow the Poisson distribution and the time it takes for the response to be fully sent follows the exponential distribution. Using the uniformization technique, MDP could be transformed into a discrete time of MDP [18]. Since the values of $\alpha(l)$ and $1/\gamma(l)$ are unknown and the number of states is high, reinforcement learning is used to obtain transition probabilities.

In Figure 3, all the possible states and actions are shown for a sample node named i that requests two layers of video. The price of the video chunk of the first and second layers is two tokens and one token, respectively. In some steps, the assumption has been made that the request sending time has finished. This assumption is made that peer i, based on its upload bandwidth, can have a maximum of 4 active request submissions for the first layer and a maximum of 3 active request submissions for the second layer at any time. Note the following points about Figure 3:

1. Time is continuous.
2. Node i, at time 1.8, cannot respond to the input request from the first layer because this peer can only have a maximum of 4 active sending requests.

In some steps, some request sending time has finished. For example, at time 3.3, a request from the first layer and a request from the second layer are finished.

3.2 Q-learning algorithm

Using the Q-learning algorithm, an optimal policy is learnt by the peers in the network. Given the optimal Q-values, $Q^*(S,a)$, the policy π^* defined by

$$\pi^*(S) = \arg\max_{a \in A(S)} Q^*(S,a) \quad (3)$$

is optimal.

To learn $Q^*(S,a)$, the value function is updated on a transition from state s to s' under action a in time $\tau_{ss'}(a)$ is as follows [19]:

$$Q^{k+1}(s,a) = Q^k(s,a) + \alpha_k(s)\left[\frac{1-e^{-\beta\tau_{ss'}}}{\beta} r(s,s',a) + e^{-\beta\tau_{ss'}} \max_{a' \in A(s')} Q^k(s',a') - Q^k(s,a)\right] \quad (4)$$

where $\alpha_k(s) \in (0,1]$ is the step size or learning rate, k is an integer variable in index successive updates, and $\beta > 0$ is chosen to be sufficiently close to 0 so that the discounted problem is equivalent to the average reward problem. α_k initially has a small value, and in the next steps, this amount is divided by the number of times that the s state is visited. It is assumed that at any moment only one event occurs, which is a basic assumption. According to the theorem proven in [20], it is supposed that $\sum_{t=0}^{\infty} \gamma_t(s,a) = a$ and $\sum_{t=0}^{\infty} \gamma_t^2(s,a) < \infty$; then for all $s \subset S, a \in A(s)$ $< \infty$, and for each pair of state-action that is infinitely updated, $Q_t(s,a)$ with probability of 1, tends to be $Q^*(S,a)$ [20].

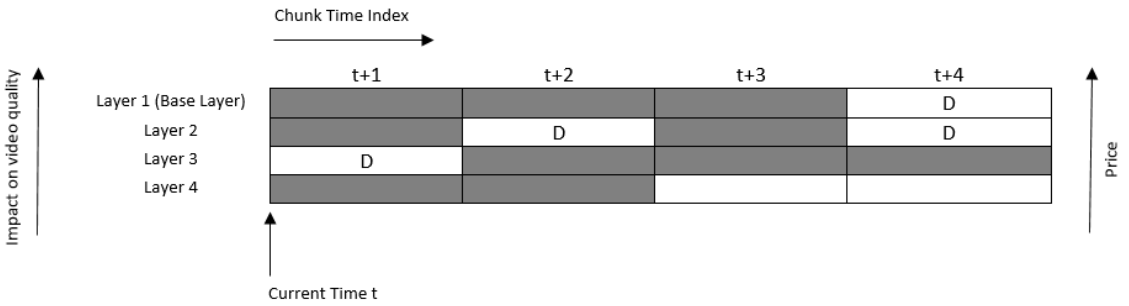

Figure 1. Buffer map in time step t.

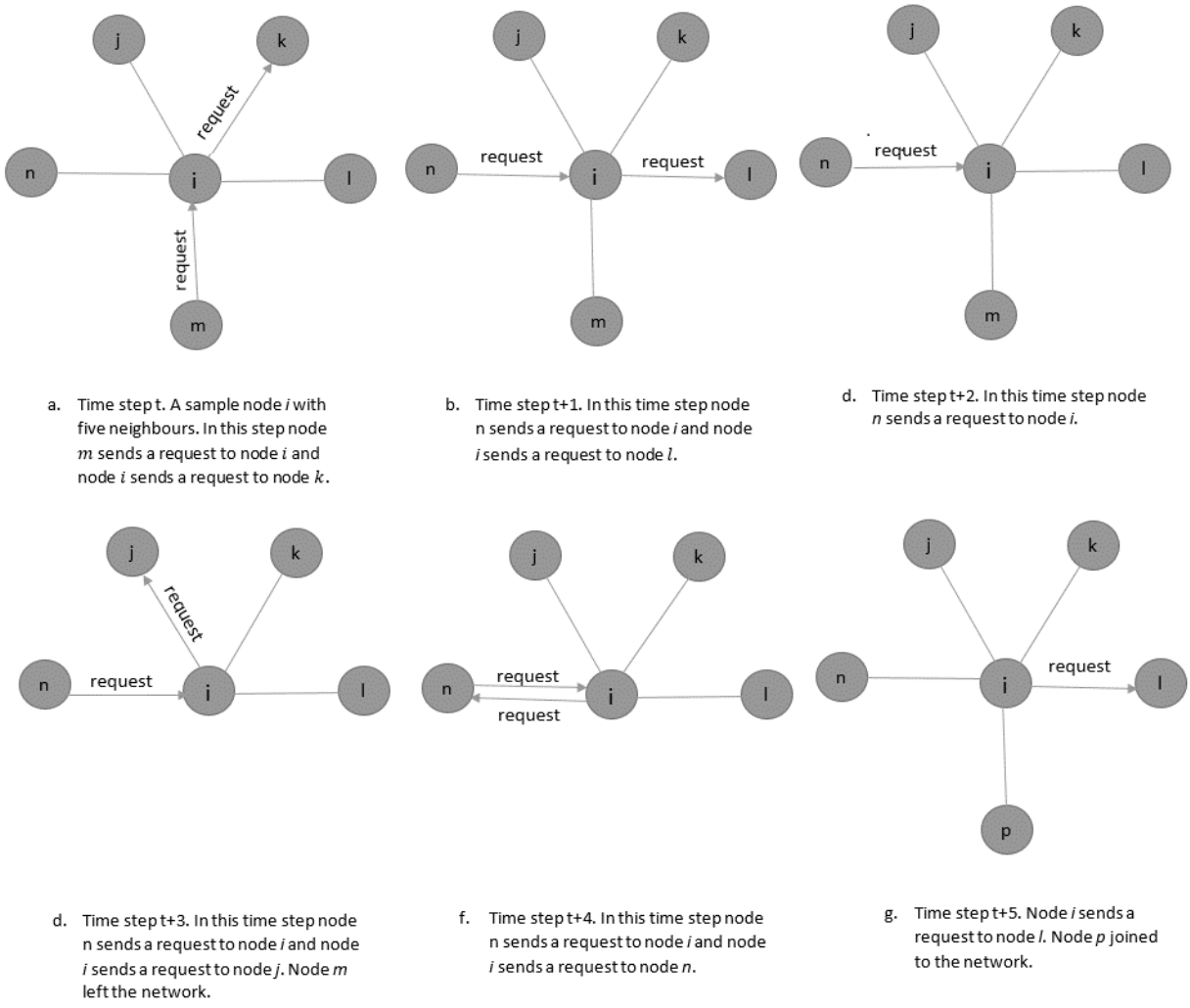

Figure 2. A sample node and its neighbors [14].

The next question is: if the Q values are not updated at the time of sending the request, will there be any changes in the optimal policy? If this is done, the space that is required to store Qs and the amount of computing will be reduced by almost one-third (the input requests and the output requests for nodes are roughly equal. Thus the completion of sending requests will be equal to the number of input requests).

Therefore, at the time of receiving or sending a request, a decision must be made, and in the meantime, one or more requests may be completed. In [21], it has been shown that if an update at the end of the sending of a request has not been made, it will not make any changes to the policy.

As shown in [20], the condition of convergence of this method is that all the state-action pairs are updated infinitely in time. Therefore, it should be possible to provide an appropriate search method for this problem.

If the *e-greedy method* [22] is used for searching the states, some states will be visited much more than the others, and therefore, the results of Q will be far from the actual values. Accordingly, the *e-directed exploration method* has been used in the proposed method to search the state space with some modifications. In fact, the continuous-time model assumed for the incoming requests causes a long time for convergence. Therefore, in the e-directed method, instead of choosing the maximum Q, the states that have the least visits are selected after some time has passed.

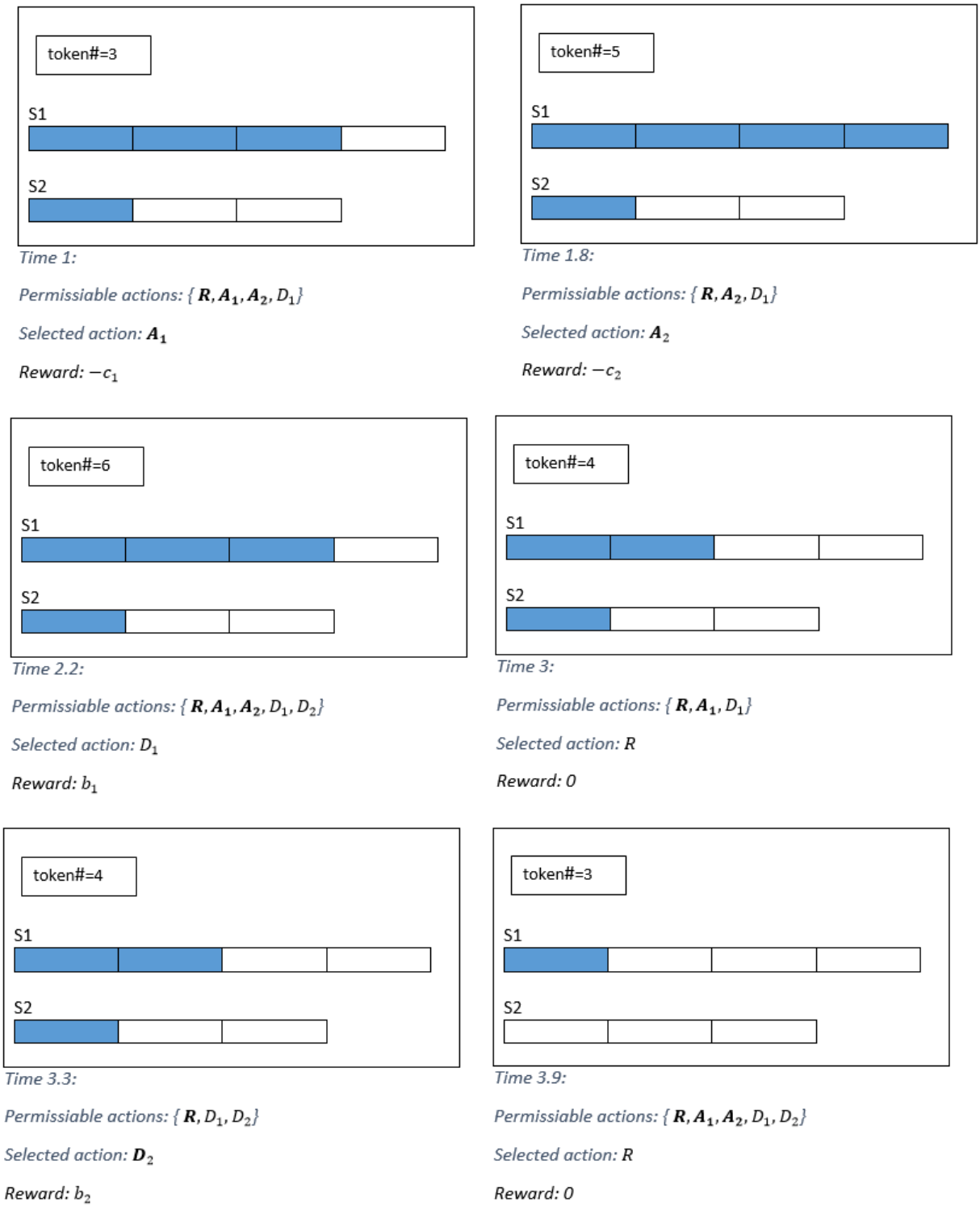

Figure 3. All possible states and actions for peer requesting two layers of video.

Another main issue in the continuous MDP formulation is the large dimensionality of the problem, i.e. the rapid growth of the state of space with the enlargement of the dimensions of the problem. It is clear that when the number of action-state pairs grows, the display of the lookup table will be impossible. Hence, it is necessary to provide a representation in which *Q* is obtained as a function of a smaller set of parameters using an approximate function.

Since the number of problem states increases with the increase in the number of layers and the increase in the size of the bandwidth, the space approximation of the states is used here.

The partition of the state space S is considered as the disjoint subsets, S_1, S_2, … , S_M, and an M-dimensional parameter vector φ is introduced, while the mth component implies to approximate the Q value function for all states $s \in S_m$ under action a. In other words, the piecewise constant approximation is being dealt with as:

$$Q(s,a,\varphi) = \varphi(m,a), if\ s \in S_m \quad (5)$$

When the value of M is small, a lookup table can be used for the aggregated problem which shows $\varphi(m,a)$ instead of $Q(s,a)$. In this work, the approximation function is used because the dimension of the problem is significant. In this case, it can be seen that the Q-learning algorithm converges to the optimal policy for the aggregated problem [21].

3.3 Proposed Q-LVS algorithm

Finally, the proposed Q-Learning Video Streaming (Q-LVS) algorithm is explained in this section. First, before running the video streaming algorithm, we will discuss the learning of optimal policy since we do not have the parameters such as the Poisson distribution, exponential distribution, and the benefit and the cost per video chunk. Then we run the algorithm of learning with different parameters. If the dimensions of the problem are small, the Q values are stored in the lookup table but if these values are large and it is practically impossible to use the lookup table, the approximation function is used and the values are saved. As mentioned earlier, the approximation function is used since the dimension of the problem is significant. The lookup table is referred to according to the parameters of the problem during each step of the execution, and the optimal policy is applied.

Figure 4 shows the pseudo-code of the proposed Q-LVS algorithm. In the initializing step, all $Q_0(s,a)$ are initialized to zero. α and β that are the parameters of Equation 4 were set to a number between zero and one. Also ∈ that is a parameter of the e-directed exploration method was set to a number between zero and one. The step iterator parameter was set to zero. In each loop parameters such as current state, nextstate, current action, available bandwidth, holding token number, and vector were calculated, and finally, Q_{n+1} based on Equation 4 was calculated.

The problem contains a feature that should be considered at the time of implementation, and this problem is that the peer's input requests occur at regular intervals, and the peer's output requests occur at continuous times. This means that each peer checks at specific intervals whether it has a request for a specific video chunk and whether its neighbours have that video chunk or not. In either case, $\tau_{ss'}(a)$ is the time between two consecutive input requests or the time between two successive output requests or one input request, and one output request or vice versa.

Initializing

$Q_0(s,a) = 0, \quad \forall_{s,a}$

Set $\alpha, \beta, \varepsilon$ a number between 0 and 1

$n = 0$

Loop

1. Observe current system state.
2. Submit current request profile if exists to P2P system.
3. Make admission decision
4. Send video chunks to requesting peers.
5. Compute $\tau_{ss'}(a)$.
6. Update system state:
   Update bandwidth, holding token number and vector s.
7. Update $Q_{n+1}(s,a)$ using (4)

Figure 4. Pseudo-Code of proposed Q-LVS algorithm.

4. Simulation

The h.264 [23-24] standard and ffmpeg [25] were used to generate bit streams in the simulation. Then the information about bit streams such as each layer frame lengths was extracted, and the other steps of simulation were done in the Matlab and the ns-2. The video rates and resolution were 0.96 Mbps and 640 * 360, respectively, and the data was 16-bit. The video was played at 30 frames per second, while the buffer map size was 8. Additionally, the sequence was encoded into three layers. The network consisted of 1000 nodes, and the bandwidth of each individual peer was equal to 2 Mbps, with each peer having approximately five neighbours. If a peer has any tokens, it randomly sends a request at each interval to one of its neighbours that has the desired video chunk, and based on its strategy, each peer may accept that request. Peer 1 acted as a server. The structure of the network is variable, meaning that the nodes may exit and enter the network during execution.

First, in the simulation section, the assumption that the incoming requests following the Poisson distribution are close to reality was examined. To do this, the input requests of a node in an array was saved, and a value of one was entered if there was a request, and a value of zero would be entered if there was no request. Next, the distance between the two input requests was calculated and saved in a new array, and finally, the values obtained with a Poisson distribution were compared. The result can be seen in Figure 5. As

it can be seen, this distribution follows a good approximation of the Poisson distribution.

In the following, the Q-LVS algorithm is compared with the other three algorithms. The following is a brief description of the three algorithms: 1) Buffer map threshold algorithm: In this algorithm, the number of video chunks of the buffer map is the basis for the acceptance or rejection of the request. If this number is more than 3, the request recipient peer will accept the request; otherwise, the peer would reject the request. 2) Bandwidth threshold algorithm:

In this algorithm, the amount of upload bandwidth available to each node is the basis for accepting or rejecting the request. If at least 20% of its upload bandwidth is free, it will accept the request; otherwise, it will reject the request. 3) Random algorithm: In this algorithm, each peer will randomly accept the incoming requests.

Figure 6 shows the upload-to-download ratio. As expected, the value of this parameter increases over time. This means that as time goes on, this rate increases. As it can be seen in the figure, the Q-LVS algorithm has a higher upload-to-download ratio than the other three algorithms. Achieving this is an ideal goal for the peer-to-peer networks since this means that the peers will upload as much as they download. This shows that the motivational algorithm works well.

In Figure 7, the efficiency of Q-LVS, bandwidth threshold, buffer map threshold, and random algorithms are displayed and compared. The efficiency is defined as the number of requests answered relative to the total number of requests [14]. As expected, the efficiency of the Q-LVS algorithm is higher than the other three algorithms, and is about 78%.

In the remainder of the simulation, the percentage of lost video was examined. This parameter shows how the performance of the Q-LVS algorithm affects the video quality. This parameter shows the percentage of video chunks that were not received successively. This percentage relates to two parameters: first, the status of the network and simulation environment, and secondly, the performance of the Q-LVS algorithm.

In Figure 8, the percentage of lost video chunks according to the video layer parameter is shown. This case was examined in two ways: the lossless channel and the 5% lossy channel. There is a high probability of losing video chunks of higher layers for two reasons: First, the number of lower layer video chunks is greater than the number of higher layer video chunks on the network. Secondly, a peer may have received a video chunk from the higher layer but since it does not have a lower layer, that video chunk will be considered lost. As expected, in a 5% lossy channel, the percentage of lost video chunks is higher.

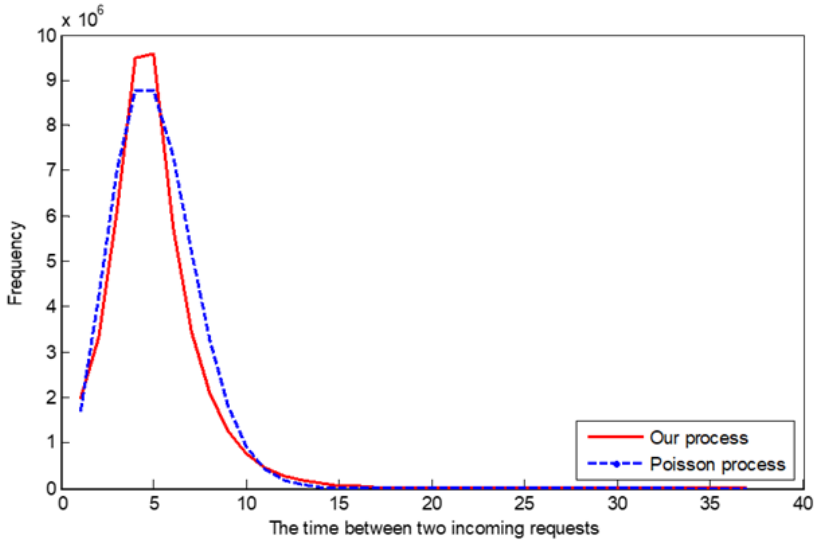

Figure 5. Comparison of distribution of incoming requests with Poisson distribution.

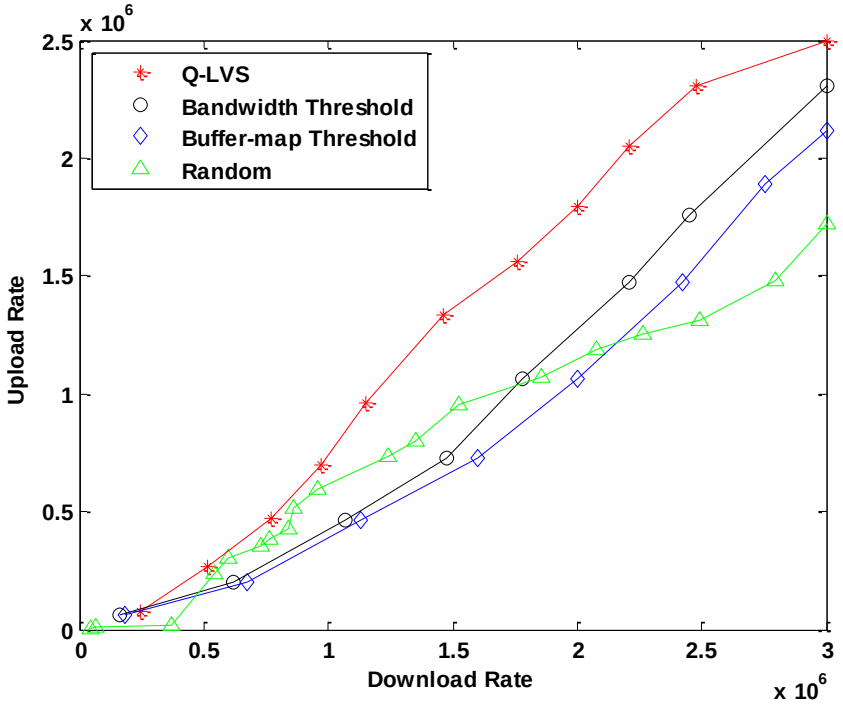

Figure 6. Upload/download ratio.

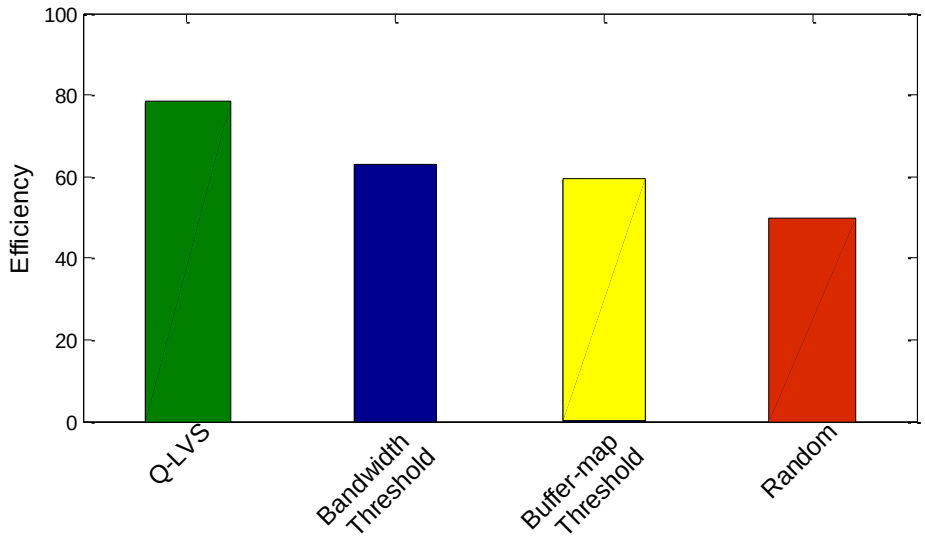

Figure 7. Performance of Q-LVS, bandwidth threshold, buffer map threshold, and random algorithms.

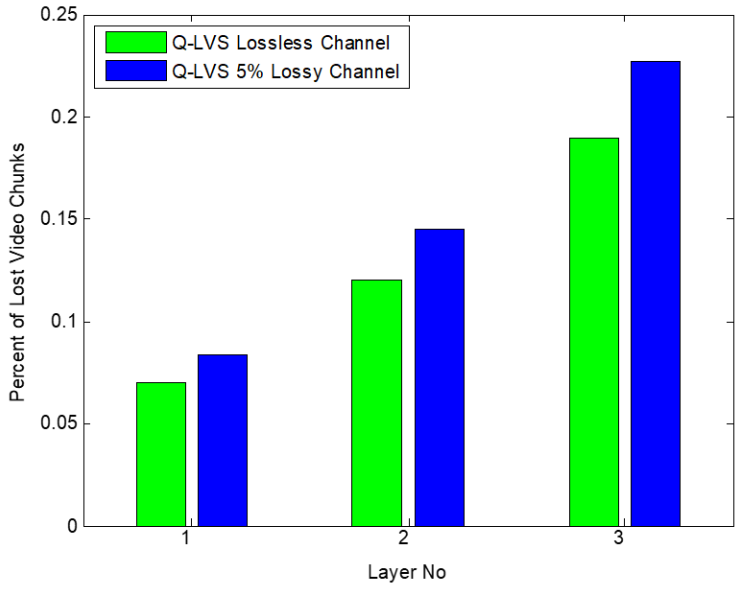

Figure 8. Percentage of lost video chunks according to video layer parameter

5. Conclusion

In this paper a token-based method was proposed aiming to incentivize the peers in a video streaming application in a mesh-based peer-to-peer network. In order to exchange video chunks, a token was assigned to all the layers, and the price of each layer was known with the video chunks of the lower layers having a higher price.

Based on the randomness of the environment, the following variables are required to be considered in the implementation, so that the optimal policy is found: the probability of receiving requests, the probability of sending requests, the amount of benefit a peer acquires from watching a video chunk, the cost incurred by sending a video chunk, the size of available bandwidth, and the length of time it takes the peer to send requests. To do this, and to take the dynamic nature of the environment into account, the continuous MDP was used. In reality, there was no statistical knowledge of the issues mentioned above, so a model-free learning algorithm was presented that could receive real-time feedback from environment dynamics and could learn the optimal policy. The QLVS algorithm was introduced, which learned the optimal policy of the peers. The simulation results show the efficiency of the algorithm.

A simulation of the proposed algorithm using deep learning and incentivizing cooperation in the context of other applications or other networks could be proposed as a future work.

References

[1] R. Schollmeier, "A definition of peer-to-peer networking for the classification of peer-to-peer architectures and applications", *In Proceedings First International Conference on Peer-to-Peer Computing*, IEEE, pp. 101-102, 2001.

[2] M. Karakaya, I. Korpeoglu, and Ö. Ulusoy, "Free riding in peer-to-peer networks", *IEEE Internet computing*, Vol. 13, No. 2, pp. 92-98, 2009.

[3] S. Saroiu, P. K. Gummadi, and S. D. Gribble. "Measurement study of peer-to-peer file sharing systems", *In Multimedia Computing and Networking* 2002, Vol. 4673, pp. 156-170. International Society for Optics and Photonics, 2001.

[4] G. Koloniari and A. Sifaleras. "Game-theoretic approaches in cloud and P2P networks: issues and challenges", *Operational Research in the Digital Era–ICT Challenges*, pp. 11-22, 2019.

[5] Lin, W. Sabrina, H. Vicky Zhao, and KJ Ray Liu. W. S. Lin, H. Vicky Zhao, and K. J. Ray Liu, "A game theoretic framework for incentive-based peer-to-peer live-streaming social networks", *In 2008 IEEE International Conference on Acoustics*, Speech and Signal Processing, pp. 2141-2144. IEEE, 2008.

[6] W. S. Lin, H. Vicky Zhao, and K. J. Ray Liu, "Incentive cooperation strategies for peer-to-peer live multimedia streaming social networks", *IEEE transactions on multimedia* 11, No. 3, pp. 396-412, 2009.

[7] Y. Zhang, and M. van der Schaar. "Designing incentives for P2P multimedia sharing", *In 2011 IEEE Global Telecommunications Conference-GLOBECOM*

2011, pp. 1-6, IEEE, 2011.

[8] V. Pai and A. E. Mohr, "Improving robustness of peer-to-peer streaming with incentives", 1st NetEcon, 2006.

[9] X. Xiao, Q. Zhang, Y. Shi and Y. Gao, "How much to share: a repeated game model for peer-to-peer streaming under service differentiation incentives", *IEEE Transactions on Parallel and Distributed Systems* 23, No. 2, pp. 288-295, 2011.

[10] A. Habib and J. Chuang, "Service differentiated peer selection: an incentive mechanism for peer-to-peer media streaming", *IEEE Transactions on Multimedia* 8, No. 3, pp. 610-621, 2006.

[11] Sh. Babazadeh, M., H. R. Navidi, H. Haj Seyed Javadi, and M. HosseinZadeh, "A New Incentive Mechanism to Detect and Restrict Sybil Nodes in P2P File-Sharing Networks with a Heterogeneous Bandwidth", *Journal of AI and Data Mining* 8, No. 4, pp. 557-571, 2020.

[12] E. Maani, Ch. Zhaofu, and A. K. K atsaggelos. "A game theoretic approach to video streaming over peer-to-peer networks", *Signal processing: image communication* 27, No. 5, 545-554, 2012.

[13] R. Aslani, V. Hakami, and M. Dehghan, "A token-based incentive mechanism for video streaming applications in peer-to-peer networks", *Multimedia Tools and Applications* 77, No. 12, pp. 14625-14653, 2018.

[14] Z. ImaniMehr, and M. DehghanTakhtFooladi, "Token-based incentive mechanism for peer-to-peer video streaming networks", *The Journal of Supercomputing* 75, No. 10, pp. 6612-6631, 2019.

[15] B. Alotibi, N. Alarifi, M. Abdulghani, and L. Altoaimy, "Overcoming free-riding behavior in peer-to-peer networks using points system approach", *Procedia Computer Science* 151, pp. 1060-1065, 2019.

[16] H. Kurdi, A. Althnian, M. Abdulghani, and S. Alkharji, "An Adjusted Free-Market-Inspired Approach to Mitigate Free-Riding Behavior in Peer-to-Peer Fog Computing", *Electronics* 9, No. 12, p. 2027, 2020.

[17] J. R. Douceur, "The sybil attack", *In International workshop on peer-to-peer systems*, pp. 251-260. Springer, Berlin, Heidelberg, 2002.

[18] P. Marbach, “Simulation-based methods for Markov decision processes”, Citeseer, 1998.

[19] S. J. Bradtke, and M. O. Duff, "Reinforcement Learning Methods for Continuous-Time Markov Decision", *Advances in Neural Information Processing Systems* 7 7, p. 393, 1995.

[20] D. P. Bertsekas and J. N. Tsitsiklis, “Neuro-dynamic programming”, *Athena Scientific*, 1996.

[21] H. Tong and T. X. Brown, "Reinforcement learning for call admission control and routing under quality of service constraints in multimedia networks." *Machine Learning* 49, No. 2, pp. 111-139, 2002.

[22] C. J. C. H. Watkins, "Learning from delayed rewards", 1989.

[23] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H. 264/AVC video coding standard", *IEEE Transactions on circuits and systems for video technology* 13, No. 7, pp.560-576, 2003.

[24] H. Schwarz, D. Marpe, and T. Wiegand, "Overview of the scalable video coding extension of the H. 264/AVC standard", *IEEE Transactions on circuits and systems for video technology* 17, No. 9, pp. 1103-1120, 2007.

[25] http://ffmpeg.org/.

Q-LVS: یک الگوریتم مبتنی بر یادگیری Q برای جریان‌سازی ویدئو در شبکه‌های نظیر به نظیر با در نظر گرفتن مکانیزم تشویقی مبتنی بر توکن

زهرا ایمانی مهر*

گروه آموزشی مهندسی کامپیوتر، دانشگاه صنعتی امیرکبیر، تهران، ایران.

ارسال ۲۰۲۱/۱۰/۱۲؛ بازنگری ۲۰۲۲/۰۲/۱۹؛ پذیرش ۲۰۲۲/۰۴/۰۶

چکیده:

در سال‌های اخیر، جریان سازی نظیر به نظیر ویدئو، مورد توجه زیادی قرار گرفته است. جریان‌سازی ویدئو به دلیل مقیاس‌پذیری بالا، کیفیت بالای ویدئو و نیاز به پهنای باند پایین، یک روش مناسب برای جریان‌سازی ویدئو در اینترنت می‌باشد. در این مقاله موضوع جریان‌سازی ویدئو در شبکه‌های نظیر به نظیر که نظیرهای خودخواه دارند، بررسی شده است. برای ایجاد انگیزه همکاری در توزیع ویدئو، از توکن به عنوان پول داخلی استفاده شده است. هر نظیر با قبول درخواست بقیه نظیرها و بارگذاری قطعات ویدئو به آنها، توکن کسب می‌کند و با ارسال درخواست و دانلود قطعات ویدئو از بقیه نظیرها توکن پرداخت می‌کند. از آنجایی که نظیرها پهنای باند متفاوت دارند، فرض شده است که ویدئو به صورت لایه‌ای کدگذاری شده است. هر نظیر بر اساس تعداد توکن‌هایش می‌تواند کیفیت درخواستی متفاوتی داشته باشد. یک مکانیزم تشویقی مبتنی بر توکن که خط مشی کنترل پذیرش نظیرها را بر اساس متغیرهای تصادفی چون پهنای باند در دسترس نظیرها، فرآیند تصادفی درخواست‌های ورودی، تقاضای نظیر برای دریافت قطعه ویدئو مورد نظر و مدت زمانی که طول می‌کشد نظیر درخواست‌ها را پاسخ بگوید، پیشنهاد شده است. درخواست‌های نظیرها در هر زمانی وارد می‌شوند بنابراین فرآیند تصمیم‌گیری پیوسته مارکوف استفاده شده است.

کلمات کلیدی: شبکه های نظیر به نظیر، کدگذاری لایه ای ویدئو، توکن، انگیزش، یادگیری ، فرآیند تصمیم گیری پیوسته مارکوف.