

A Proposed Peer Selection Algorithm for Transmission Scheduling in P2P-VOD Systems

Hatem Fetoh

Department of Information
Technology
Faculty of Computers and
Information
Mansoura University, Egypt

Waleed M. Bahgat

Department of Information
Technology
Faculty of Computers and
Information
Mansoura University, Egypt

Ahmed Atwan

Department of Information
Technology
Faculty of Computers and
Information
Mansoura University, Egypt

Abstract—Video transmission in peer-to-peer video-on-demand faces some challenges. These challenges include long transmission delay and poor quality of service. The peer selection plays an important role in enhancing transmission efficiency. For this reason, a proposed algorithm for peer selection is introduced to overcome these challenges. The proposed algorithm consists of four steps. First, the peers exchange their own buffer maps with other peers. Second, the requested segments are ordered according to their priorities. Third, neighbors of the receiver are evaluated by the efficiency estimation. Finally, the efficient sender list is applied to solve the overloading and bottleneck on the highest efficient sender. A simulation is introduced to evaluate the performance of the proposed algorithm compared to a peer selection algorithm with context-aware adaptive (CAA) data scheduling algorithm. The results show that, the proposed algorithm reduces initial buffering delay and achieves high throughput rather than CAA algorithm.

Keywords—video-on-demand; segment transmission; chunk miss ratio; initial playback delay; peer-to-peer

I. INTRODUCTION

In P2P environment, there is no dedicated server. A peer can play as server and client at the same time. [1] The overall performance of P2P is much better than client server. There are many reasons for that. First, the peers in the network communicate and share bandwidth with each other. Second, any peer can join or leave the network dynamically. Finally, each peer serves other peers by sharing its upload bandwidth. This reduces the bandwidth costs of the server. [2]

The receiver peer must receive unavailable segments before playback deadlines. This leads to start-up buffering delay reduction. Available segments are cached in the peer's buffer. Then, peers can exchange these segments with other peers. [3]

To exchange these segments, there are two main approaches that are push-based approach and pull-based approach. In push-based approach, a sender peer pushes the available segments to a receiver. In tree-based overlays, the parent peer uses push-based approach to send available segments to children peers. In pull-based systems, each peer maintains a buffer map that is exchanged and updated periodically in P2P system. Mesh-based overlays use pull-based approaches. [4] Application examples for mesh-based overlays are PPlive [5], Coolstreaming [6].

Transmission scheduling in P2P-VOD relies on the buffer maps (BMs) exchange among peers in the network. Buffer map contains available segments, unavailable segments and playback point for each peer. In the P2P mesh-based network, the transmission scheduling algorithm is used to obtain a transmission schedule from each sender. By a transmission schedule, the sequence of requested segments is specified, their senders are selected and their transmission times for each segment are estimated. In the reason of the time constraints on the segment transmission for P2P-VOD, a transmission scheduling is a difficult task.

There are some drawbacks in the previous transmission scheduling algorithms. For random scheduling algorithm [7], it has long buffering delay to interest a video content. Overloading of fast peers is a drawback of rarest first algorithm [3]. The On-time Delivery of VBR streams (ODV) algorithm ignores the segment's playback deadline time. [3] For context-aware adaptive data scheduling algorithm (CAA), [8] the drawback is that the delivery time to pull segment from neighbors to receiver is relatively high.

There are some challenges that face Video-on-Demand in P2P. These challenges include long playback delay and the service quality. This paper proposes an algorithm for peer selection to overcome these challenges.

The proposed algorithm has two main contributions. First, the reputation factor is introduced to measure the response of each sender. Second, the efficiency of each sender is estimated to evaluate the senders and select the most efficient sender. As a result, the packet loss during transmission is reduced since the next sender is selected from the efficient sender list instead of using the first sender. This is alternative solution that is better than using the first sender since it is affected by many problems such as overloading and network bottleneck.

The remainder content of this paper is organized as follows. First, the previous transmission scheduling algorithms are discussed in section II. Second, the paper includes the efficient peer selection algorithm in section III. Third, this algorithm is evaluated by using experimental results in section IV. Finally, the conclusion is introduced in section V.

II. RELATED WORK

P2P VOD systems' users suffer from some problems. These problems are initial buffering delays and play out lags

[9]. This reduces the video quality, increases server bandwidth costs. Hence, the previous researches suggest the heuristic transmission scheduling algorithms in peer-to-peer streaming systems such as [6], [10], and [11]. The objective of transmission scheduling algorithms is the performance improvement in P2P-VOD systems. In the following section, the most important previous segment scheduling algorithms are presented.

A. Round-Robin Algorithm (RR)

In this algorithm, each requested segment is received sequentially from list of senders. Each sender has the same probability to send a number of segments. This algorithm doesn't rely on the bandwidth factor or other factors to select the sender.

The main advantage of this algorithm is high balancing of transmission load to all senders. Unfortunately, the drawback of this algorithm is that it leads to non-optimal sender for transmission and causes an inefficient transmission scheduling. [8]

B. Random Pull scheduling Algorithm (RP)

In this algorithm, a receiver requests the segment from more peers. The requests segments are determined through a window sliding. Selection of the segments and their senders is a random selection.

This algorithm has some drawbacks. The drawbacks are long buffering delay, high packet loss and low throughput. [7]

C. Rarest first algorithm (RF)

Some applications use RF algorithm such as CoolStreaming/DONet. There are three main steps for this algorithm. First, the partners' count for each requested segment is estimated. Second, the segments with fewer partners are more difficult to meet deadline constraints. This leads to quickly transmission before deadline time. Hence, requested segments are ordered by partners' count factor for each segment. In the case of that there is one sender only, it must be selected. However, there are a number of partners for the requested segment; the algorithm selects the sender with highest bandwidth.

Unfortunately, this algorithm has some drawbacks. First, there are long buffering delay. Second, unavailable segments are transmitted from the highest speed senders. This causes overloading on these senders. [3]

D. On-time Delivery of VBR streams (ODV)

There are three main steps for ODV algorithm. First, buffer map is exchanged among peers to obtain the information needed. In addition, the expected transmission time from each sender to a receiver is computed. Finally, the selected sender is the sender can send the segment in the earliest time. [3]

Unfortunately, it has some drawbacks. First, initial buffering delay is relatively high. Second, the priority estimation of the segments ignores the playback deadline times.

E. Context-aware Adaptive Scheduling Algorithm (CAA)

CAA consists of four steps. First, the segment's priority is estimated for each requested segment from a receiver. Second, it estimates the bandwidth of each sender. Third, it estimates the expected delivery time that is needed to send a requested segment from sender to receiver. Finally, the sender with the earliest delivery time is selected. [8]

Unfortunately, CAA has a drawback. The initial buffering delay to stream a video content is still high.

F. An efficient hybrid push-pull based Protocol

In this protocol, it consists of two phases that are push-based approach and pull-based approach. The video segments are classified into urgent and non-urgent segments. First, urgent segments are transmitted by push-based approach. Second, non-urgent segments are transmitted by pull-based approach. [12]

The drawback of this protocol is that it did not discuss the sender selection approach relying on any factor. This leads to inefficient video transmission in the reason of long playback buffering delay and high chunk miss ratio.

The best one from these transmission scheduling algorithms is CAA. The reason of this is enhancing of the bandwidth estimation. This leads to delivery time reduction from the selected senders to a receiver and the throughput performance improvement.

III. PROPOSED ALGORITHM

To overcome the bottleneck on the selected sender in the previous algorithms, the efficient peer selection algorithm (EPS) is introduced. It estimates the efficiency of the senders relying on the historical reputation values in the last cycles. The proposed efficient peer selection algorithm is described in terms of four steps. These steps are the same in the proposed framework that is shown in Fig .1 [13]. First, peers exchange buffer maps over the network. Second, the requested segments are scheduled by the priority estimation of each segment. Third, neighbors are evaluated by the efficiency estimation of each neighbor peer. Fourth, the efficient senders list is applied to order the senders by the efficiency. The highest efficiency sender is selected for the transmission of requested segment to the receiver peer. TABLE .1 shows the used parameters in the efficient peer selection algorithm (EPS).

TABLE I. PARAMETERS OF THE PROPOSED ALGORITHM

Parameter	Value
y	Requested segment
P_Id	Receiver's playback point
BM [x][y]	Availability of segment y in peer x
Pri[y]	Priority of the requested segment y
Neighview [x]	Neighbours set of peer x
dC	The segment with max. priority
Segset	Requested segments' set
S _{ac}	The selected sender of the segment dC
Eff (I)	The efficiency of peer I
Maxeff	The highest efficiency of all senders
S	Selected sender

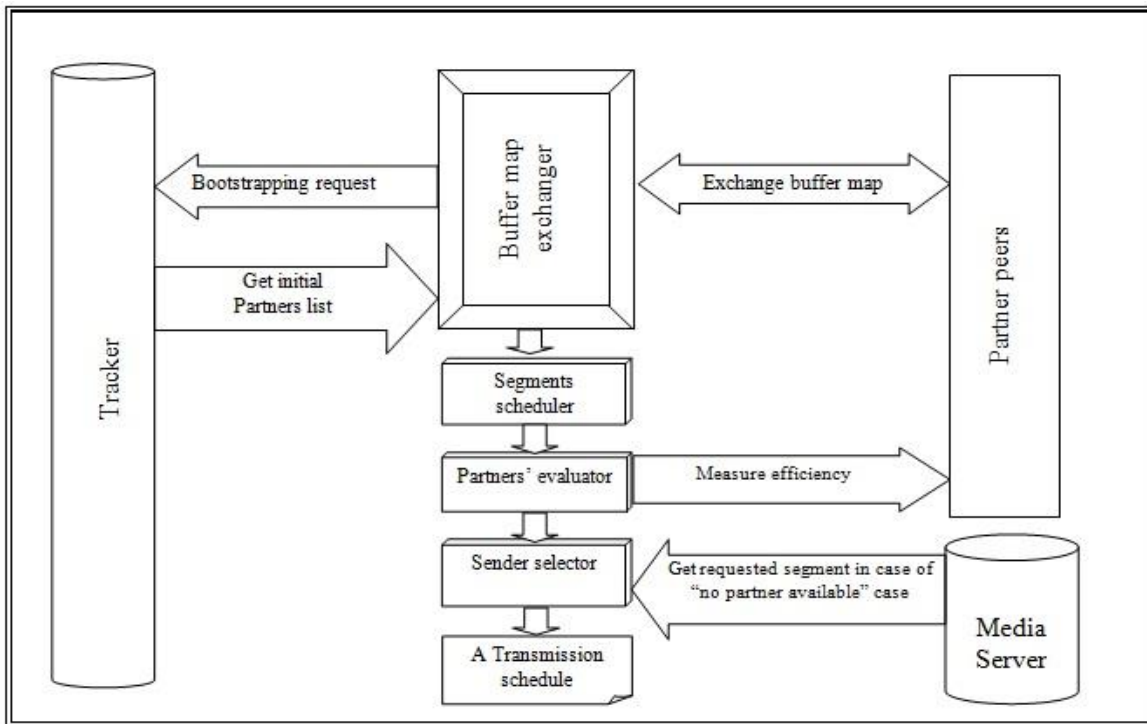


Fig. 1. The proposed framework of the transmission segment scheduling of P2P-VOD.[13]

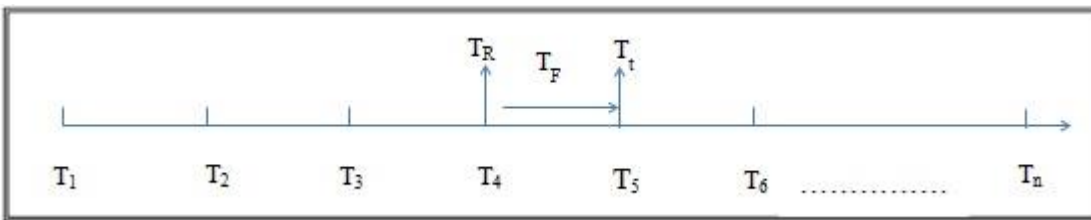


Fig. 2. Case 1 in the segment scheduler algorithm

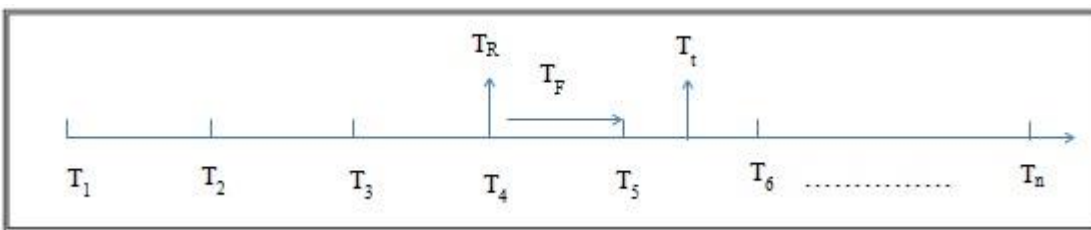


Fig. 3. Case 2 in the segment scheduler algorithm

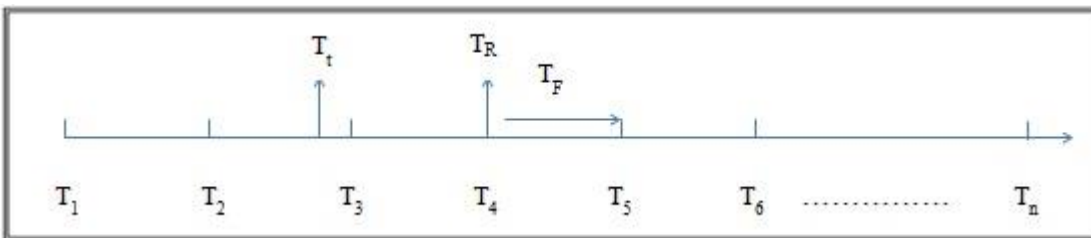


Fig. 4. Case 3 in the segment scheduler algorithm

The requested segments are determined by the buffer map of the receiver. Requested segments are ordered by the priority estimation. The requested segment's priority is estimated by a

segment scheduler algorithm. The segment with the highest probability for missing is the highest priority in case 2. The parameter T_R is the playback point of the receiver. The highest

priority segment must be transmitted quickly in the time Threshold T_t . The requested segment is transferred in the time T_f that is estimated as $T_f = \text{size}(y) / B_y$; the parameter B_y represents the bandwidth of y . In the case 1, it's shown in Fig .2, if T_t equals $(T_R + T_F)$ this means that the segment must be transmitted now. The priority of this segment is the highest. In the case 2, it's shown in Fig .3, if T_t is more than $(T_R + T_F)$, the priority is estimated by the equation (1), and this means that the highest priority segment is the one that has the highest probability to be missed. In the case 3, it's shown in Fig .4, if T_t is lower than $(T_R + T_F)$, this means that this segment is missed, priority equals zero.

Segment scheduler algorithm (SCA)

```

If  $T_t = T_R + T_F$  Then Pri[y] = Max. // Case 1
Else If  $T_t > T_R + T_F$  Then // Case 2
Pri[y] =  $1 / [T_t - (T_R + T_F)]$  (1)
Else // Case 3
Pri[y] = 0
End if
    
```

The proposed efficient peer selection algorithm estimates the reputation of each peer that acts as a neighbor to the receiver. First, it counts the requested segments (N) which are requested from the sender I in the time cycle T_n . this algorithm relies on the historical reputation information. Second, it estimates the number of non-transmitted segments or the number of segments that are transmitted after their playback deadline times (M). Then, it estimates the reputation of the sender i in the time T_n by the equation (2).

$$R_{T_n}(I) = \frac{N - M}{N} \quad (2)$$

Finally, it estimates the efficiency of the sender by the equation (3) that is relied on historical reputation values in the last cycles ($T_1, T_2, T_3, \dots, T_{k-1}$).

$$Eff_I[j][k] = \sum_{j=1}^m \lambda_j \cdot R_{T_j}(I)[j][k-1] \quad (3)$$

The objective of EPS algorithm is selecting the most efficient sender from a large number of senders. To evaluate each sender, the efficiency parameter is introduced by the equation (3). The traditional average method is not used to estimate the efficiency of the senders according to the reputation values. The reason of this is that the weights of the time cycles are different. In the case of that the reputation value of the last cycle k-1 is low but the first cycle is very high, the average of two values gives good efficiency. The highest probability of the reputation value of the next cycle is near from the last cycle k-1. The effect of the historical reputation values of the last cycles is higher than the old cycles. Hence, the weight of the reputation in the last cycle k-1 is higher than or equals the cycle k-2 and the same in the other cycles. For the traditional average method, it leads to inefficient sender selection for transmission scheduling in P2P-VOD system. To avoid an inaccurate efficiency estimation problem, the proposed approach is introduces in equation (3). In this equation, the parameter m is a number

more than zero, λ_j is a constant, $0 \leq \lambda_j \leq 1$, $\sum_{j=1}^m \lambda_j = 1$

$\lambda_j \geq \lambda_{j+1} \geq \lambda_{j+2} \geq \dots \geq \lambda_m$. The algorithm estimates the efficiency of each sender I in the next cycle k. afterwards, it selects the sender with the highest $Eff_I[j][k]$ to be an efficient sender for the requested segment. This is because that it improves throughput and can transmit segments in the earliest transmission time. This leads to selection of this peer to be the sender of the requested segment.

The proposed efficient peer selection algorithm introduces efficient senders list (ESL). This list contains the senders with high efficiency score that are ordered by the efficiency score in descend order. Then, it considers the first sender in this list as the most efficient one to be selected. In addition, if the highest sender in the list (ESL) is affected by any problem such as overloading or instability, this leads to transmission failure. The proposed algorithm solves this problem by the next sender selection in (ESL) instead of the first sender.

Algorithm(EPS) : Efficient Peer Selection Algorithm

```

(1) For each  $y \in \text{SlideWindow}$ 
(2)   And  $y > P\_Id$  and  $BM[x][y] = 0$  do
(3)     segSet = segSet  $\cup$  {y} // set of requested segments
(4)     Pri[y] = Es_Pri (y) //compute priority by SCA algorithm
(5)   End for y
(6)   For each  $i \in \text{Neighview}[x]$  do
(7)     Eff[i] = Comp_Eff (i) //estimate efficiency acc. (3)
(8)   End for i
(9)   While SegSet  $\neq$  null do
(10)    dC  $\leftarrow$  argy { Max Pri[y] ,  $y \in \text{SegSet}$  }
(11)    Sdc = { $\phi$ }
(12)   For each  $i \in \text{Neighview}[m]$  do
(13)     If Eff (i) > Maxeff (S) Then {
(14)       MaxOSS (S) = OSS(i)
(15)       Sdc = i
(16)     }
(17)   End for P
(18) SegSet = SegSet - {dC}
    
```

IV. EXPERIMENTAL RESULTS

In order to evaluate the proposed efficient peer selection algorithm, Opnet Modeler 14.5 simulator is used. [14] Number of nodes in this simulation is 1500. The parameters of this simulation are shown in TABLE .2. A Dell server is used in simulation that has memory is 8GB, CPU is 8 cores Intel Pentium, and the operating system is RedHat AS5. These are the same parameters that are used in the environment of CAA algorithm.

TABLE II. PARAMETERS EVALUATION TABLE

Parameter	Value	Unit
Initial Bandwidth Between Peers	10	Mbps
Numbers Of Neighbours	30	
Peer Nodes	1500	
Tracker Server I	1	
Tracker Server Bandwidth I Gbps	1	Gbps

Average throughput of CAA versus EPS under no. of peers

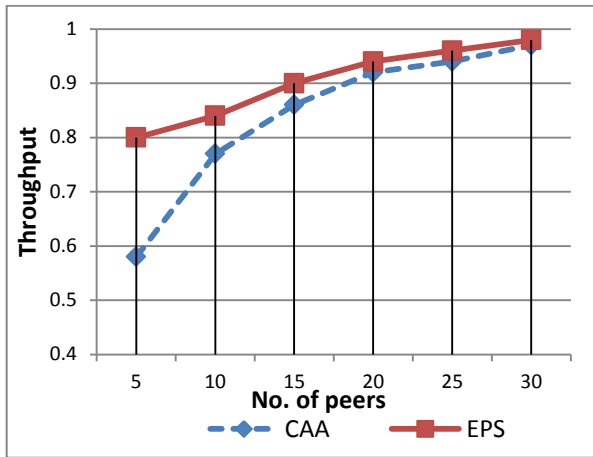


Fig. 5. Average throughput of proposed algorithm versus CAA under no. of peers

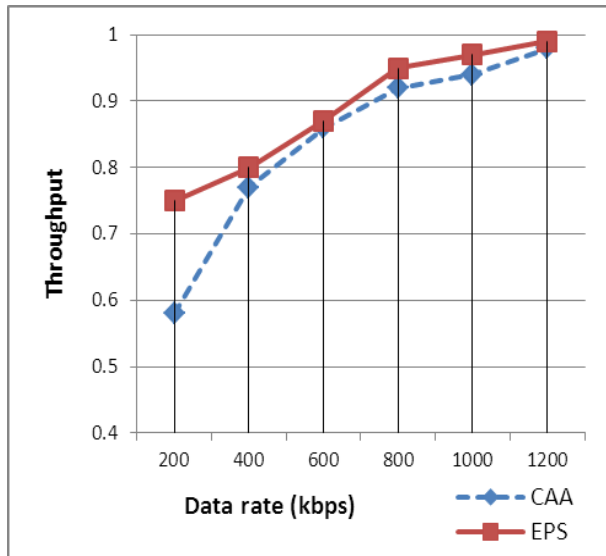


Fig. 6. Average throughput of proposed algorithm and CAA under data rate

In the first experiment, the peers exchange buffer map from each one to the others. The results are demonstrated in Fig .5. This figure shows the average throughput performance of the efficient peer selection algorithm versus CAA at number of peers (5 – 30) peer. In Fig .5, the results show the following:

- Efficient peer selection algorithm outperforms CAA under different peers' number.
- For both algorithms, in the case of that peers' number is increased, this improves the average throughput performance. This is because that the probability of high efficient senders becomes high.
- In the range (5-10), the throughput is more degraded than the other cycles. This is due to the low probability to search for an efficient peer in this region. This result indicates that the throughput of EFS is higher than CAA. The reason is that the selection of high responded

peers with high reputation scores leads to improve the efficiency of the transmission from multi-senders to a receiver. This leads to throughput performance improvement.

Average throughput of CAA algorithm versus EPS under data rate

The range of data rate values are (200 – 1200) kbps. In Fig .6, these results indicates the following:

- EFS algorithm outperforms CAA under the range of data rate values (200-1200).
- For EPS algorithm, prediction of the high efficient senders leads to improvement of the throughput performance. Throughput is nearly constant in the range (800-1200). The reason is that the higher efficiency senders are affected by overload problem from large number of receivers.

Average throughput of CAA algorithm versus EPS under peer sample

Random peers are selected for this experiment. The number of these peers is twenty. The results of the experiments are shown in Fig .7. The results indicate the following:

- The EPS algorithm outperforms CAA at the range of peer samples in the most peers. The reason is that the transmission relies on selection of stable peers. These peers achieved high reputation scores. This leads to high performance for EPS algorithm.

Buffering delay of CAA versus EPS algorithm

1) Impact of Peers' Number:

The results of average buffering delay of EPS algorithm and CAA at count of peers (5 – 30) are shown in Fig .8 the results indicate the followings:

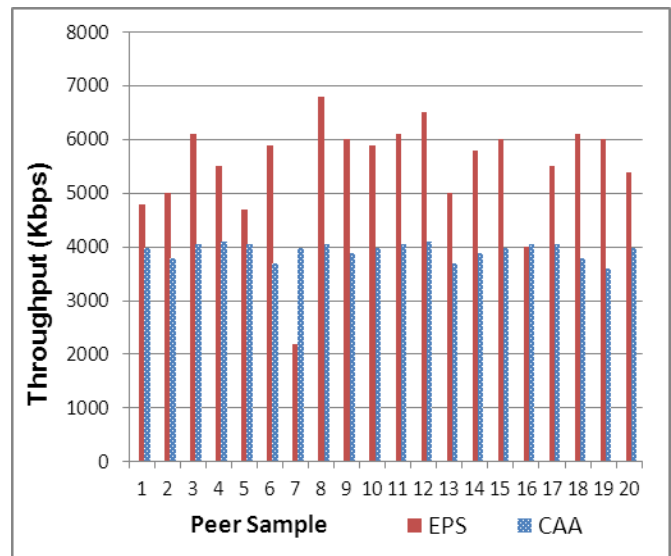


Fig. 7. Average throughput of proposed algorithm and CAA algorithm under peer sample

- The proposed EPS algorithm enhances CAA under various peers' count.

- The EPS algorithm reduces buffering delay by selecting the most efficient peer that is capable of segment transmission in the earliest time.

2) *Impact of Transmission data rate:*

The results of the buffering delay of EPS versus CAA at data rate range (200-1200) kbps are shown in Fig .9. The results indicate the followings:

- EPS equals CAA in data rate (200,600 and 800) kbps and it outperforms CAA in data rate (400, 1000 and 1200) kbps.
- EPS algorithm reduces buffering delay in the most from CAA. This is because that EPS is based on peer's efficiency score measurement and selection the best sender that transmits the requested segments in the lowest time. This leads to buffering delay reduction by EPS algorithm.

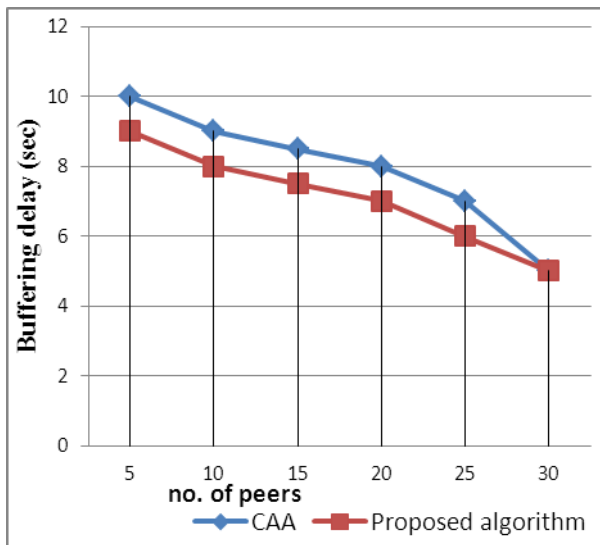


Fig. 8. buffering delay of the proposed algorithm EPS versus CAA under no. of peers

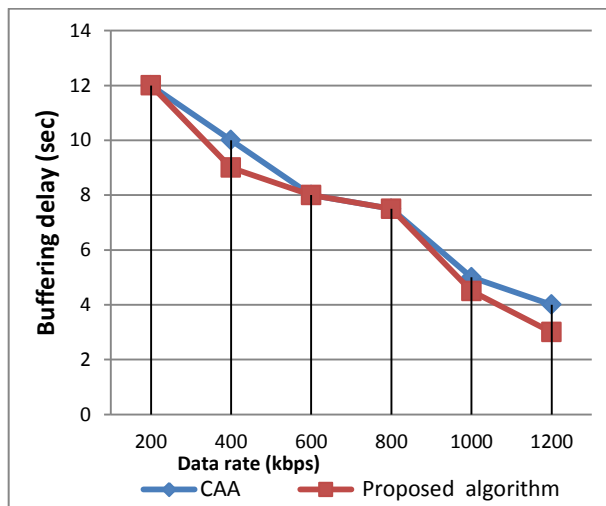


Fig. 9. buffering delay of proposed algorithm EPS versus CAA algorithm under data rate

V. CONCLUSION

This paper introduces a proposed algorithm for peer selection in P2P-VOD. The key contribution of the proposed algorithm is adding the reputation evaluation of peers in the historical time cycles.

This reputation is based on the number of transmitted segments before their deadlines segments compared with the number of requested segments from each sender. According to that, the proposed algorithm overcomes the bottleneck problem that occurs on the selected sender.

A simulation is introduced to evaluate the proposed algorithm compared to CAA algorithm. Our simulation results proved that there is a significant improvement in the performance of the proposed algorithm rather than CAA. The initial buffering playback delay of P2P-VOD system is decreased, continuity index is enhanced and average throughput is improved.

REFERENCES

- [1] García Pineda, Miguel, et al. "Controlling P2P File-Sharing Networks Traffic." *Network Protocols and Algorithms* 3.4 (2011): 54-92.
- [2] Milošević, Dejan S., et al. "Peer-to-peer computing." (2002).
- [3] Kowalski, Greg, and Mohamed Hefeeda. "Empirical analysis of multi-segment transmission algorithms in peer-to-peer streaming." *Multimedia, 2009. ISM'09. 11th IEEE International Symposium on.* IEEE, 2009.
- [4] Xu, Xiaofeng, et al. "A peer-to-peer video-on-demand system using multiple description coding and server diversity." *Image Processing, 2004. ICIP'04. 2004 International Conference on.* Vol. 3. IEEE, 2004.
- [5] Hei, Xiaojun, et al. "A measurement study of a large-scale P2P IPTV system." *Multimedia, IEEE Transactions on* 9.8 (2007): 1672-1687.
- [6] Zhang, Xinyan, et al. "CoolStreaming/DONet: a data-driven overlay network for peer-to-peer live media streaming." *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE.* Vol. 3. IEEE, 2005.
- [7] Liang, Chao, Yang Guo, and Yong Liu. "Is random scheduling sufficient in P2P video streaming?." *Distributed Computing Systems, 2008. ICDCS'08. The 28th International Conference on.* IEEE, 2008.
- [8] Li, Wei, Quan Zheng, and Song Wang. "Context-aware adaptive data scheduling algorithm for P2P streaming systems." *Consumer Electronics, Communications and Networks (CECNet), 2012 2nd International Conference on.* IEEE, 2012.
- [9] Y. Liu, Y. Guo, C. Liang, "A Survey on Peer-to-Peer Video Streaming Systems," *Peer-to-Peer Networking and Applications* vol. 1, no. , pp. 18–28, 2008.
- [10] V. Pai, K. Kumar, K. Tamilmani, V. Sambamurthy, and A. Mohr, "Chainsaw: Eliminating trees from overlay multicast," in *Proc. of International Workshop on Peer-to-Peer Systems (IPTPS'05)*, Ithaca, NY, pp. 127–140 , February 2005.
- [11] V. Agarwal and R. Rejaie, "Adaptive multi-source streaming in heterogeneous peer-to-peer networks," in *Proc. of ACM/SPIE Multimedia Computing and Networking (MMCN'05)*, San Jose, CA, pp. 13–25, January 2005.
- [12] Mohamed Ghetas , Putra Sumari, " An Efficient Hybrid Push-Pull Based Protocol for VOD In Peerto-Peer network", *IOSR Journal of Computer Engineering (IOSR-JCE)* 16.3(2014) , 67-73
- [13] Fetoh, Hatem, Waleed M. Bahgat, and Ahmed Atwan. "A proposed framework for segment transmission scheduling in P2P video streaming." *Informatics and Systems (INFOS)*, 2014.
- [14] <http://opnet-modeler.software.informer.com/14.5/>