*Journal of Computational and Theoretical Nanoscience*
Vol. 14, 5907–5914, 2017

RESEARCH ARTICLE

# Artificial Bee Colony Based Data Scheduling for Peer to Peer Network Video on Demand System

G. Sivakumar* and V. Venkatachalam

*Department of Computer Science and Engineering, Erode Sengunthar Engineering College, Erode 638057, Tamilnadu, India*

P2P media streaming applications have become popular in recent years. The nodes in the network help to relay received data to other peer nodes in the P2P media streaming systems thereby removing the bottleneck of central servers. Data scheduling, which focuses on dealing with dispatching and transmitting data segments within a system efficiently and optimally is the most important issue in designing a P2P VoD system. An efficient data scheduling based on Artificial Bee Colony (ABC) algorithm is presented in this paper. We develop the network in a hierarchical form as an extension of our previous work. The media file is cached as data segments by each peer in the network and based on the priority function these segments are prioritized. By ABC algorithm the peer with optimal uplink bandwidth is selected for scheduling. The prioritized segments are scheduled to the peer from the selected peers and further it sends the request to the proxy server. Compared to the existing work, scheduling time and throughput of our proposed approach are improved and is shown by the Simulation results.

**Keywords:** P2P VoD System, Data Scheduling, Artificial Bee Colony (ABC) Algorithm, Priority Function.

## 1. INTRODUCTION

An alternative network model provided by the traditional client-server architecture is the Peer-to-peer (P2P). A decentralized model is used by the P2P networks where each machine referred to as a peer functions as a client with its own layer of server functionality.[1–3] At the same time the role of a client and a server is played by a peer. That is, the requests can be initiated by a peer to other peers and at the same time respond to incoming requests from other peers on the network. It differs from the traditional client-server model where a client can only send requests to a server and then wait for the server's response. With a client-server approach, as the number of clients requesting services from the server increase the performance of the server will deteriorate. However, as an increasing number of peers are added to the network the overall network performance actually improves in P2P networks. At the same time each peer can upload and download and new peers can join the group while old peers leave at any time in a process like this. To the end-users this dynamic re-organization of group peer members is transparent.[4,5]

*Author to whom correspondence should be addressed.

Another characteristic of a P2P network in terms of fault-tolerance is its capability. The P2P application will continue by using other peers when a peer goes down or is disconnected from the network. For example, any clients downloading a certain file are also serving as servers in a Bit Torrent system.[6–8] A client searches for other peers, picks up parts of the file where the old peer was and continues the download process when it finds one of the peers is not responding. A P2P network is more fault-tolerant compared to a client-server model, where all communication will stop if the server is down.

In internet, the Video-on-Demand (VoD) streaming service has become extremely popular for past few years, for instance: the Internet TV, online video, distance education, etc. The P2P computing technique is credited as the most effective technique to offer the "play-as-download" VoD services in the networks with dynamic heterogeneous environment.[9,10] After thorough investigation the gossip-based P2P streaming is considered as robust and dependable in order to achieve large scale VoD services. Whenever the playback offsets known as partners are available every peer arbitrarily links with its peers in the conventional gossip based P2P network with the VoD servers.[11,12] The peers keep sharing the available data with its partners in P2P network and thus the node collects it from its partners when in need of a specific video.

The main issue to improve the effective utilization rate of the bandwidth of general peers is to enhance the capacity of the P2P VoD system. So to improve the utilization rate of the system data scheduling is the solution. In peer to peer network many data scheduling schemes have been proposed. For scalable video streaming over P2P networks Hua et al. have presented an efficient scheduling algorithm in Ref. [13]. Each block which are stored in the caches of peers are prioritized. Then by using their proposed block scheduling algorithm the maximum priority sum of video blocks was scheduled. However, scheduling time and throughput of the system are to be improved further. We present an efficient scheduling algorithm to overcome this problem. Our contribution is described as follows.

- As we have done in our previous work we develop the network in a hierarchical form.
- Video files in each peer are stored as data segments. Using the priority function we prioritize these data segments.
- We present Artificial Bee Colony (ABC) algorithm for optimal peer selection. Then from the selected peer the prioritized data segments are scheduled.

Rest of this paper is organized as follows. Some previous works are reviewed in Section 2. Our proposed Artificial Bee Colony based data scheduling is presented in Section 3. In Section 4 the Results of our proposed work are discussed. With Section 5 this paper is concluded.

RESEARCH ARTICLE

## 2. RELATED WORKS

With some previous work that has been done in the same domain we relate our projected work in this segment. A block scheduling scheme known as the P2P layered video streaming has been addressed in Hua et al.[13] In accordance with the block's significance for video playback they defined a soft priority function for each block to be requested by a node. Their proposed priority function was unique and it stroked good balance between different factors which made the priority of a block to represent the relative importance of the block over a wide variation of block size between different layers. Then to an optimization problem that maximized the priority sum of the delivered video blocks they applied block scheduling problem. Both distributed and centralized scheduling algorithms were developed for the problem.

In P2P network Data scheduling strategy is critical to make full use of node resource and it can help optimize user experience as well as system throughput in peer to peer Video on Demand system. But the efficient scheduling of media data for a normal peer still remains a challenging task. Due to the dynamic characteristics of P2P network the problem that occurs is NP-hard problem. By presenting Genetic Algorithm based data scheduling Huang et al.[14] have overcome the problems. By applying four steps of genetic algorithm such as selection, cross over, mutation and evaluation they have optimized the scheduling effect.

Hierarchically Clustered P2P Video Streaming has been proposed by Guo et al.[15] The streaming rate approaching the optimal upper bound while accommodating large viewer population is supported by their proposed streaming framework. The peers are grouped into clusters and formed a hierarchy among them. To balance the bandwidth resources across clusters and to maximize the supportable streaming rate the Peers were assigned to appropriate cluster. Retrieving data chunks from source and neighboring peers and utilizing its uplink bandwidth to serve data chunks to other peers by presenting distributed queue-based scheduling algorithms were determined. Peers' uplink bandwidths are fully utilized and also the streaming rate close to the optimum in practical network environment is supported by their proposed queue-based scheduling algorithms.

A discrete and slotted mathematical model to analyze chunk selection algorithms has been introduced by Chen et al.[16] Rarest first algorithm and greedy algorithm has also been included. A performance metric, the optimization function for evaluating chunk selection algorithms and for the exploration of chunk dissemination strategies respectively has also been presented. The use of peer resources has been promoted and using their proposed service request randomization mechanism chunk requests from rendezvous on a few of peers has been prevented. By employing weight assignment strategies they also avoided excessive requests for rare chunks.

Simple and effective hybrid chunk fetching strategy called Threshold Bipolar(TB), which partitioned the buffer of a peer into head and tail parts and employed different chunk download strategies for the two parts has been proposed by Li et al.[17] Using the selfish strategies peer downloaded the chunks in the head part depending on the TB strategy. If the head part was fully filled it used altruistic fetching strategies to download chunks in the tail part. To characterize the buffer progresses of peers in startup stages they also developed analytical models.

A cooperative scheduling mechanism with a two-level topology designed to work on large-scale distributed computing P2P systems has been proposed by Rius et al.[18] Three criteria which only uses local information to schedule tasks thus providing scalability to the overall scheduling system has been proposed. By setting up these three criteria the system easily adapted to work efficiently with different kinds of distributed applications. The importance of good scheduling in such heterogeneous systems is justified by the extensive experimentation which is carried out and also emphasized the importance of having a scheduling algorithm capable of being adapted to the requirements of different kinds of application.

Xia et al.[19] have introduced an evolved layered P2P (E-LP2P) data scheduling scheme in the process of service delivery to relieve the negative effect brought by the intricate wireless network environment and unstable user behavior in layered mobile peer-to peer (P2P) streaming service. According to its importance in streaming play to guarantee the basic play of streaming the data in base layer was scheduled. According to the characters of streaming data, including its position and amount in server peer set in a multiple tied way towards the data in enhancement layer the data in enhancement layer was scheduled. Jitter prevent mechanism was used to adjust the highest layer dynamically during the process of data scheduling to cope with the layer jitter caused by the fluctuation of bandwidth.

Efficient data scheduling schemes has been offered in the above reviewed literature. However, scheduling time and throughput of the system has to be improved. So we present Artificial Bee Colony optimization algorithm for optimal peer selection and data scheduling in this paper. Throughput and scheduling time of our system is improved than that of existing work and this is shown in the simulation results.

## 3. ARTIFICIAL BEE COLONY BASED DATA SCHEDULING P2P VIDEO ON DEMAND

### 3.1. Overview

We present Artificial Bee Colony (ABC) optimization algorithm for data scheduling in peer to peer video on demand system. As we presented in our previous work we develop the peer to peer network in hierarchical form. Each peer in the network caches the video as a data segments and using the priority function these segments are prioritized. A Peer sends data request to the proxy server which tracks all peers in the network if it wants video to play or download. Then to select optimal peers which cache the requested data segments in the network our proposed ABC algorithm is applied. Finally, from the selected optimal peer video segments are scheduled to the peer that requests for data. The peer directly sends request to the VoD server if requested data is not available in the neighbor peer. The block diagram of our proposed approach is shown in Figure 1.

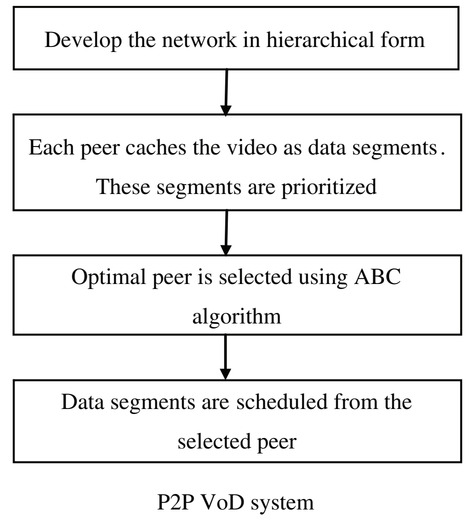


**Fig. 1.** Block diagram of our proposed approach.

### 3.2. System Model

The network is developed in a hierarchical form as an extension of our previous work. To all the nodes in the next layer the first layer of peer nodes around the VoD server is considered as the parent node. The nodes in the second layer are considered as its child nodes by the node in the first layer. The nodes in the same layer are considered as sibling of one another. Proxy is situated at the prime locations of the network. The system or network model of our proposed approach is shown in Figure 2.

Peer P2a sends request to the proxy server through its parent peer P2 in the above figure. If requested data is available the proxy server schedules the data from the parent nodes P1, P2 and P3. Likely through its parent peer P1peer P1a sends request to the proxy server. The request is directly send to the VoD server if the requested data is not available in the proxy server. Then to the peer P1a the VoD server sends the requested data.

The videos are cached as data segments by each peer in the system. Using the Eq. (1) the data segments in the cache of peer are prioritized. Playback time of each data segment is considered for prioritization. The segments have higher priority to be scheduled to the peer. The priority calculation is defined as follows:

$$P_l = 1/(t_l - t_{\text{current}}) + \delta * (1/\text{Neigh}(\text{Seg}_l)) + \text{ran}(0, 1) * \varphi \tag{1}$$

Where, $P_l$—denotes the priority of segment $l$, $t_l$—Playback time length of segment $l$, $t_{\text{current}}$—Current playback time, $\text{Neigh}(\text{Seg}_l)$—Number of neighbor peers which hold the segment $l$, $\delta$—Coefficient of local rarest first strategy which is used to adjust the priority between the rarest priority and urgent priority of segments. $\varphi$—trembling coefficient used to adjust the trembling scope when two segments have the same rarest priority.

The proxy server schedules the requested data from the other peer nodes if a peer sends data request to the proxy server via parent node. Optimal peer is selected among the peers in the network using Artificial Bee Colony algorithm for data scheduling. Service capabilities of each peer are considered as the input attributes to the ABC algorithm for peer selection. The attributes taken as input to the ABC algorithm are the Uplink bandwidth, Euclidean distance and online time of the peer. Service capability of the neighbor peer $n$ of peer $m$ is described as follows

$$C_m^n = [t_m^n * (b_m^n)^2]/d_m^n \tag{2}$$

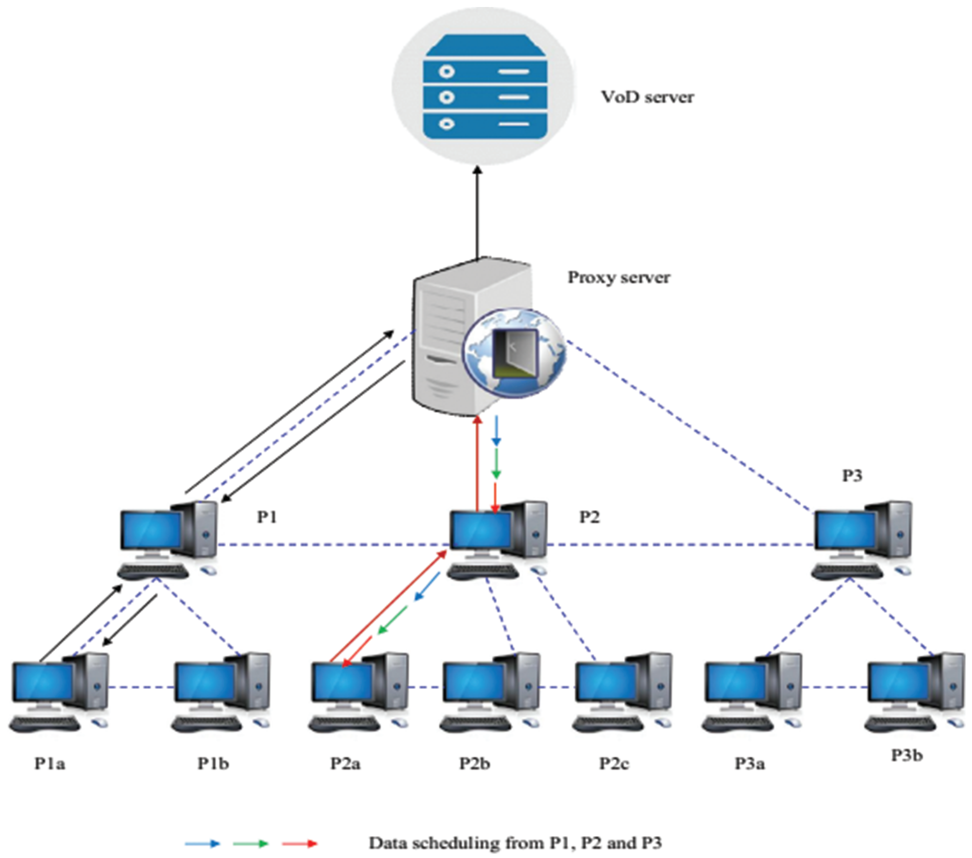


**Fig. 2.** System model.

Where, $t_m^n$—denotes the online time of the source/parent peer $n$ of peer $m$, $b_m^n$—denotes the uplink bandwidth of source/parent peer $n$ of peer $m$ $d_m^n$—denotes the Euclidean distance between source/parent peer $n$ of peer $m$.

### 3.3. Artificial Bee Colony Based Data Scheduling

The colony of artificial bees contains three groups of bees: employed bees, onlookers and scouts in the ABC algorithm. An onlooker is known as a bee waiting on the dance area for making decision to choose a food source and an employed bee is a bee going to the food source visited by it previously. A bee carrying out random search is called a scout. First half of the colony consists of employed artificial bees and the second half constitutes the onlookers in the ABC algorithm. There is only one employed bee for every food source. In other words, the number of employed bees is equal to the number of food sources around the hive. The employed bee whose food source is exhausted by the employed and onlooker bees becomes a scout.

#### *3.3.1. Initialization Phase*

We initialize $x_i$ food sources which are known as solutions of our work in this phase. Each solution is D dimensional vector. D is the number of optimization parameters. We have three optimization parameters that are online time, bandwidth and distance in our work. The food sources are initialized as follows.

| | | | | | |
|---|---|---|---|---|---|
| For $i=1$, | $(t_m^n)^1$ | $(b_m^n)^1$ | $(d_m^n)^1$ | (or) | $(C_m^n)^1$ |
| For $i=2$, | $(t_m^n)^2$ | $(b_m^n)^2$ | $(d_m^n)^2$ | (or) | $(C_m^n)^2$ |
| | | ⋮ | | | ⋮ |
| For $i=N$,(or) | $(t_m^n)^N$ | $(b_m^n)^N$ | $(d_m^n)^N$ | | $(C_m^n)^N$ |

Where, $N$ represents the population size of each iteration. Then the fitness function is calculated as follows.

$$F_i = \begin{cases} \dfrac{1}{1+f_i} & \text{if } f_i \geq 0 \\ 1+ab(f_i) & \text{if } f_i < 0 \end{cases} \tag{3}$$

Where, $f_i$ denotes objective function value of food sources and based on the service capability of peersit is calculated. It is defined as follows

$$f_i = \sum_{n \in \text{Neigh}_m,\, s_{mn}^k = 1} C_m^n \tag{4}$$

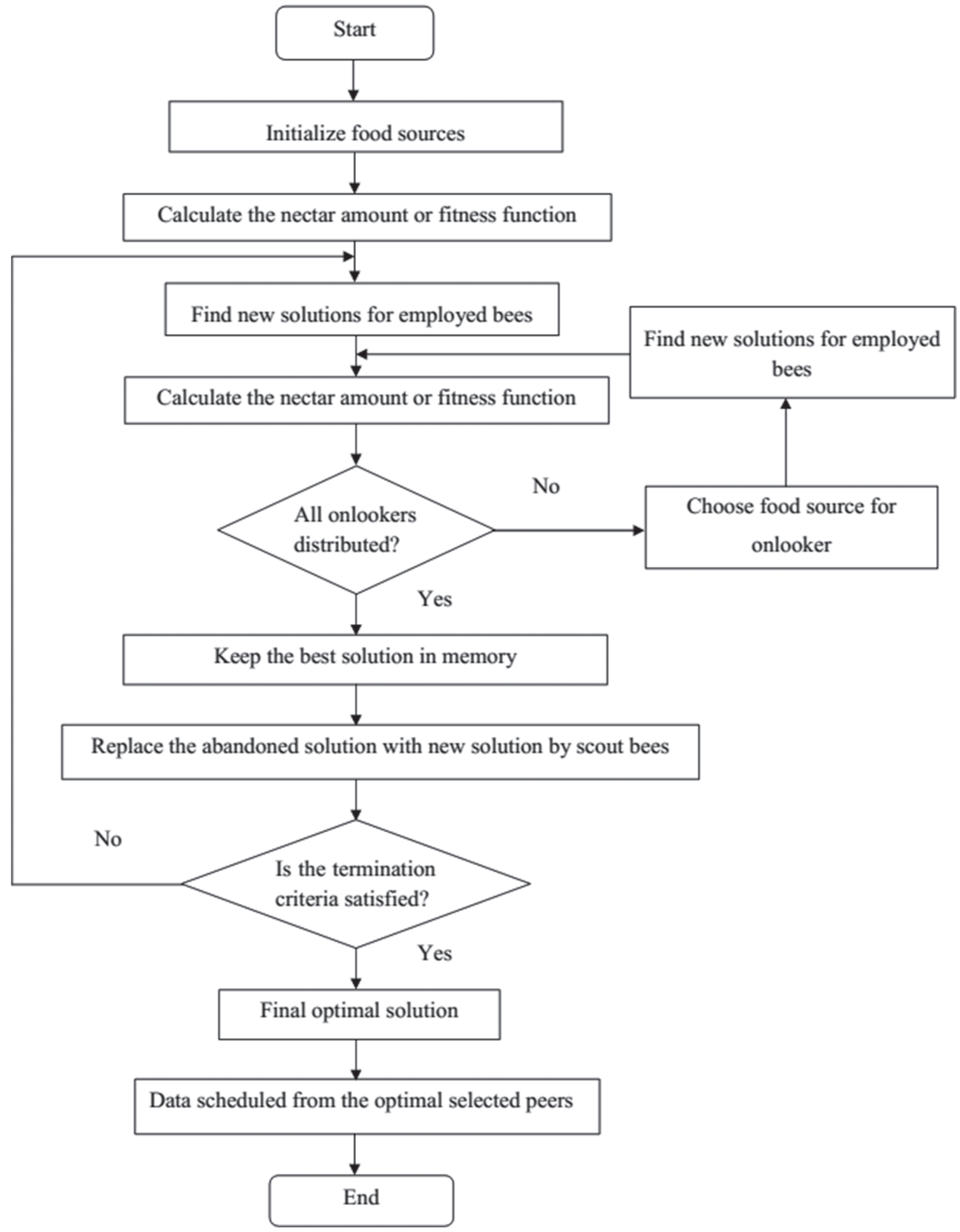


**Fig. 3.** Flowchart for algorithm.

Where, $s_{mn}^{k} = 1$ represents the data segment $k$ requested by peer $m$ is stored at peer $n$, while $s_{mn}^{k} = 0$ represents data segment $k$ is not at peer $n$. Maximum fitness value is selected as the quality of the best food source.

### *3.3.2. Employed Bees Phase*

Every employed bee goes to the food source area visited by her at the previous cycle since that food source exists in her memory, and then chooses a new food source by means of visual information in the neighborhood of the present one in employed bees phase. The new solution or Neighbor food source of $x_i$ is determined and it is calculated as follows

$$v_{ij} = x_{ij} + \phi_{ij}(x_{ij} - x_{lj}) \tag{5}$$

Where, $j$ represents a randomly selected parameter index, $x_l$ represents a randomly selected food source, $\phi_{ij}$ represents a random number within the range $[-1, 1]$.

Then using Eq. (3) the nectar amount or fitness value of new source is determined. Greedy selection is applied between $x_i$ and $v_i$ after the calculation of fitness value of new source. The bee memorizes the new position and forgets the old one if the nectar amount of the new source is higher than that of the previous one. Otherwise she keeps the position of the previous one. They share the nectar information of the food sources (solutions) and their position information with the onlooker bees on the dance area after all employed bees complete the search process.

RESEARCH ARTICLE

#### *3.3.3. Onlooker Bee Phase*

The nectar information taken from all employed bees are evaluated by an onlooker bee and depending on the probability value associated with that food source $p_i$ calculated by the following expression it chooses a food source

$$p_i = \frac{F_i}{\sum_{i=1}^{N} F_i} \tag{6}$$

Onlooker bee produces a modification on the position (solution) using Eq. (5) in her memory and checks the nectar amount of the candidate source (solution) as in the case of the employed bee. The bee memorizes the new position and forgets the old one providing that its nectar is higher than that of the previous one.

#### *3.3.4. Scout Phase*

The scout bees search the new solutions randomly. By the scouts the food source whose nectar is abandoned by the bees is replaced with a new food source. This is simulated by randomly producing a position and replacing it with the abandoned one in the ABC algorithm. If a position cannot be improved further through a predetermined number of cycles then that food source is assumed to be abandoned in the ABC algorithm.

Until the maximum number of cycles (MNC) the above phases are continued. The requested data segment is scheduled from the selected peer after getting the optimal solution which is corresponding to the optimal neighbor peer. For example, suppose if the selected peer nodes are indexed like [2, 1, 3, 2, 1, 2], it means that the data segment 1 is selected from peer 2, data segment 1 from peer 1 and so on. The peer directly sends request to the VoD server if requested data is not available in the neighbor peer.

**Algorithm.**

(1) Initialize the food sources.
(2) Calculate the fitness function of food sources using (3)
(3) Find new solutions for employed bees using (5)
(4) Apply greedy between previous and new solutions
(5) Select the visible solutions for onlooker bees using (6)
(6) Find new solutions for onlooker bees using (5)
(7) Apply greedy between previous and new solutions
(8) Replace the abandoned solutions if they exist with the new solutions using scout bee.
(9) Keep the best solution so far.
(10) Repeat until reach the maximum number of cycles.
(11) From the selected optimal peer, data segment is scheduled.

## 4. RESULTS AND DISCUSSION

Our proposed Artificial Bee Colony based network is simulated using the network simulator NS2. We have used the BitTorrent packet-level simulator for P2P networks. In this work we have used peer nodes and these nodes are performing in the region of 1000 m × 1000 m. Table I shows the simulation parameters of our proposed approach. The media file "Grandmother" is examined in our proposed approach. Size of this media file 4345 kb and it is segmented with size of 4 kb approximately. Each peer in the network caches these segmented files. Simulation topology of our proposed approach is shown in Figure 4. In this figure, node 0 and 2 denote the VoD server and proxy server respectively. Peers 1, 5 and 3 are the parent nodes of the network. As shown in Figure 5, Peer 21 sends the video request for the video "Grandmother" to the proxy server through its parent peer 1. The proxy server tracks the all peer in the network. The peer with efficient uplink bandwidth is selected using the ABC algorithm. The data segments with higher priority are scheduled from the selected

**Table I.** Simulation parameters.

| Parameters | Value |
|---|---|
| No. of nodes | 28 |
| Area | 1000 × 1000 |
| Link bandwidth | 2 Mb |
| No. of receivers | 2 |
| Traffic source | CBR |
| Traffic rate | 100 kb–500 kb |
| Packet size | 1000 bytes |
| Simulation time | 50 secs |

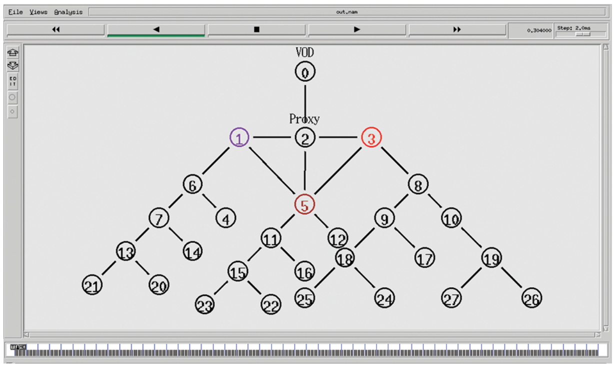


**Fig. 4.** Simulation topology.

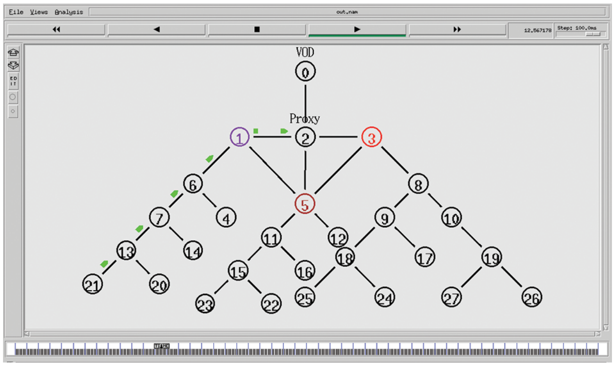


**Fig. 5.** Peer 21 sends request to the proxy.

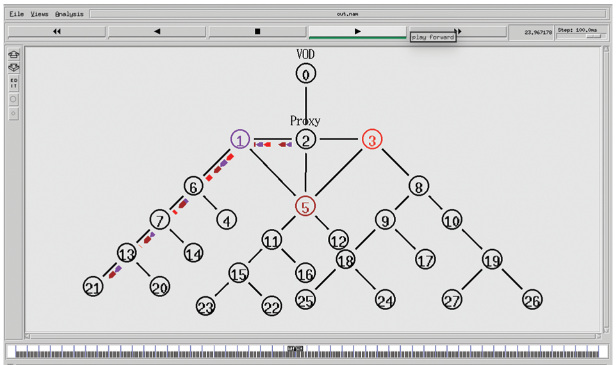


**Fig. 6.** Data is scheduling to the peer 21.

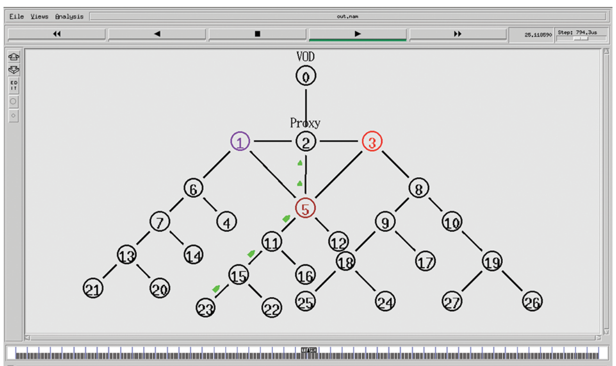


**Fig. 7.** Peer 23 sends request to the proxy.

optimal peers as shown in Figure 6. Likely peer sends video request to the proxy server through its parent peer as shown in Figure 7. The requested data is not available in the peer so the request is directly sent to the VoD server as shown in Figure 8. The VoD server sends response with the data to the peer as shown in Figure 9. The whole simulation is done with the time period of 50 seconds.

### 4.1. Performance Based on Rates

In this section, performance metrics of our proposed approach are evaluated for varying rates 100, 200, 300, 400 and 500 kb. Figures 10–13 show that delay, throughput,

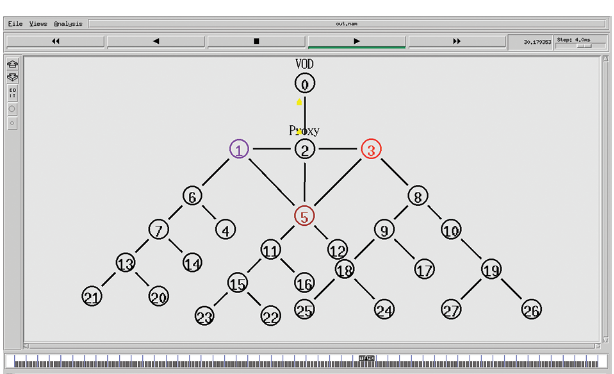


**Fig. 8.** Proxy forwards the received request to VoD server.

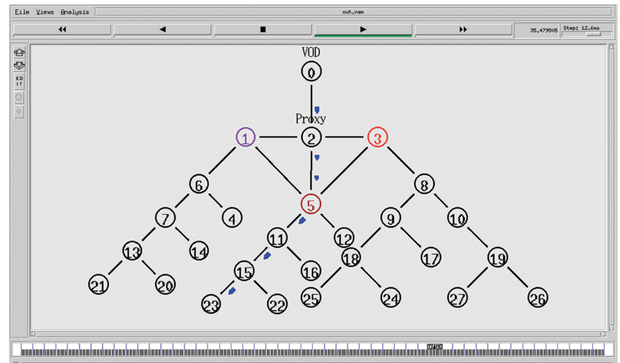


**Fig. 9.** VoD server forwards data to the peer 23.

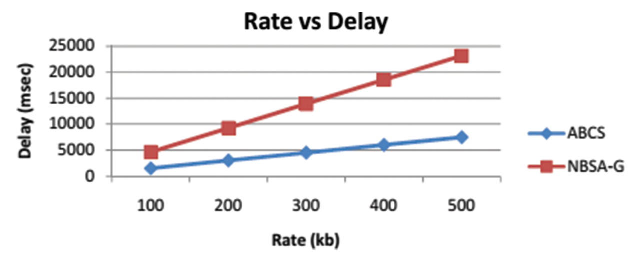


**Fig. 10.** Rate versus delay.

delivery ratio and bandwidth of our proposed approach for varying rates. Performance metrics of our proposed approach Artificial Bee Colony based data scheduling (ABCS) are compared with that of existing work NBSA-G.[13] Figure 10 shows the delay of our proposed approach for varying rates. When the rate increases, scheduling time of the system is also increased. But compared to the existing work, delay of our proposed approach is reduced to 68%. Because of the optimal peer selection using ABC algorithm, scheduling time of our system is reduced. Throughput of our proposed approach is shown in Figure 11. In our work, we have scheduled only the prioritized data segments from the optimal peers so that throughput of our proposed approach is increased to 94% than that of the existing work. Figure 12 shows the delivery ratio of our proposed approach. Compared to the existing work, delivery ratio of our proposed approach is increased to 98%. Bandwidth utilization of our proposed approach is shown in Figure 13. When the number of rate increases, bandwidth utilization of the system is decreased. But compared to the existing work, bandwidth utilization of our proposed approach is increased to 18%.

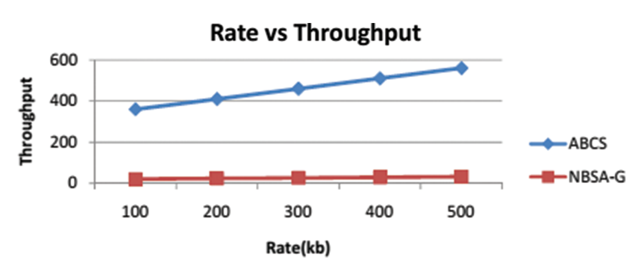


**Fig. 11.** Rate versus throughput.

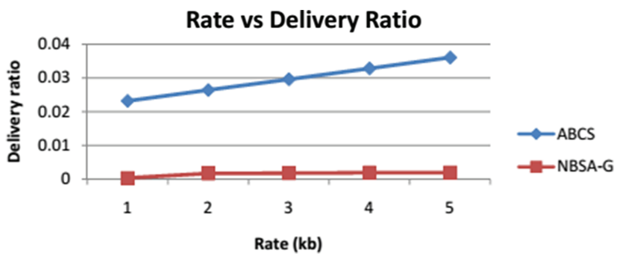


**Fig. 12.** Rate versus delivery ratio.

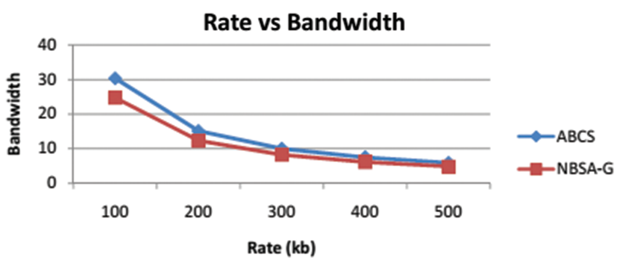


**Fig. 13.** Rate versus bandwidth.

## 5. CONCLUSION

In this paper, we have presented Artificial Bee Colony based data scheduling in P2P VoD system and the work has been simulated using the network simulator. In this work, we have structured our network in the form of hierarchical as we done in our previous work. We have stored the media file as a data segments in the peer. These data segments were prioritized using the priority function. After receiving the video request from a peer, ABC algorithm has been applied to select the optimal peer. Then the prioritized data segments were scheduled from the optimal peers. Simulation results showed that scheduling time and throughput of our proposed approach were improved that of the existing work.

## References


1. K. K. Ramachandran and B. Sikdar, *IEEE Transactions on Dependable and Secure Computing* 8, 617 (**2011**).
2. Q. He, J. Yan, Y. Yang, R. Kowalczyk, and H. Jin, *IEEE Transactions on Services Computing* 6, 64 (**2013**).
3. S. K. Awasthi and Y. N. Singh, *IEEE Communications Letters* 20, 1345 (**2016**).
4. C. Zhao, J. Zhao, X. Lin, and C. Wu, *IEEE/ACM Transactions on Networking* 24, 2607 (**2016**).
5. J. S. Leu, M. C. Yu, and H. C. Yueh, *Journal of Network and Systems Management* 23, 803 (**2015**).
6. M. Hawa, J. S. Rahhal, and D. I. Abu-Al-Nadi, *Peer-to-Peer Networking and Applications* 5, 279 (**2012**).
7. J. Funasaka, Dynamic piece uploading for initial seeding method on bittorrent-like P2P systems, *First International Symposium on Computing and Networking (CANDAR)*, IEEE (**2013**), DOI: 10.1109/CANDAR.2013.83.
8. N. M. Sarband, M. R. Khayyambashi, and N. Movahedinia, *Peer-to-Peer Networking and Applications* 10, 1051 (**2017**).
9. A. Nafaa, B. Gourdin, and L. Murphy, *Multimedia Tools and Applications* 59, 169 (**2012**).
10. C. S. Lin, *Multimedia Tools and Applications* 62, 701 (**2013**).
11. V. Slavov and P. Rao, *The VLDB Journal* 23, 51(**2014**).
12. Z. Li, J. Cao, and G. Chen, ContinuStreaming: Achieving high playback continuity of gossip-based peer-to-peer streaming, *IEEE International Symposium on Parallel and Distributed Processing*, IEEE (**2008**), DOI: 10.1109/IPDPS. 2008.4536252.
13. K. L. Hua, G. M. Chiu, H. K. Pao, and Y. C. Cheng, *Computer Networks* 57, 2856 (**2013**).
14. Y. Zhou, Y. Dai, G. Huang, P. Liu, and X. Li, Data scheduling strategy in P2P VoD system based on adaptive genetic algorithm, *Proceedings of the 2012 International Conference on Information Technology and Software Engineering*, Springer, Berlin, Heidelberg (**2013**), pp. 429–440.
15. Y. Guo, C. Liang, and Y. Liu, *Computer Networks* 56, 3432 (**2012**).
16. C. Hu, M. Chen, and C. Xing, *Computer Networks* 57, 3009 (**2013**).
17. C. Li, C. Chen, Y. Liu, and B. Zhang, *Computer Networks* 70 , 154 (**2014**).
18. J. Rius, F. Cores, and F. Solsona, *Journal of Network and Computer Applications* 36, 1620 (**2013**).
19. H. I. Xia, N. Wang, and Z. M. Zeng, *The Journal of China Universities of Posts and Telecommunications* 19, 60 (**2012**).


Received: 13 July 2017. Revised/Accepted: 18 July 2017.