

Chapter 46

Data Scheduling Strategy in P2P VoD System Based on Adaptive Genetic Algorithm

Ya Zhou, Yunping Dai, Guimin Huang, Pingshan Liu
and Xuanfeng Li

Abstract In Peer-to-Peer (P2P) Video-on-Demand (VoD) systems, data scheduling strategy is critical to make the most use of nodes in P2P network, and it also helps optimizing the time consuming of data scheduling and startup delay as well as the utilization of neighbors' uplink bandwidth. But it is a challenging task to design an efficiently scheduling strategy. On one hand, the scheduling strategy should balance the need of segments between one requesting node and the whole system; On the other hand, the problem of scheduling segment transmission is a NP-hard problem due to the dynamic characteristics of P2P network. To resolve those problems we propose a scheduling strategy based on adaptive genetic algorithm (AGA). The scheduling strategy is based on segments' emergency degree, supply and demand ratio and regional position to select the needed segments to schedule in a scheduling cycle. Then through the AGA to select the suitable supply nodes to transmit segments. Experiment results show that our proposed algorithm outperforms the RANDOM and Hill-climbing in terms of reducing the time consuming of each data scheduling, startup delay, and making the most use of the network uplink bandwidth.

Keywords Peer-to-peer · Video-on-demand · Data scheduling · Adaptive genetic algorithm · Segment selection

Y. Zhou (✉) · Y. Dai (✉) · G. Huang · P. Liu · X. Li
Research Center on Data Science and Social Computing, Guilin University of Electronic
Technology, No 1, Jinji Road, Qixing Area, Guilin City, Guang Xi, China
e-mail: ccyzhou@guet.edu.cn

Y. Dai
e-mail: 610300039@qq.com

46.1 Introduction

In recent years, Peer-to-Peer (P2P) technology has become increasingly popular, and leads to the rapid development of P2P VoD systems which achieve some noteworthy breakthroughs. Nowadays there are some successful commercial applications, such as: PPLive, QQLive and QVOD.

In P2P VoD systems, a video is usually partitioned into segments with equal size. The segments are transmitted from multiple source nodes to a requesting node. Then, the requesting node caches the requested segments and advertises those segments to the system. All those operations are determined by the scheduling strategy which is run on the requesting node. In addition, the data scheduling also imposes time constraints on segment transmission. If segments arrive at the requesting node after the playback deadline, the segments are essentially useless. So data scheduling is a main factor that impacts the user perceived visual quality and also may cause the playback failure [1].

In order to reduce the time consuming of each data scheduling and the startup delay, improve the utilization of neighbors' uplink bandwidth and make most use of the network nodes, we design an efficient data scheduling strategy for P2P VoD systems in this paper. The strategy is based on segments' emergency degree, supply and demand ratio and regional position to select the needed segments to schedule in a scheduling cycle, then through the adaptive genetic algorithm to select the suitable supply nodes to transmit segments.

The remaining parts of this paper are structured as follows. In Sect. 46.2, we discuss the related works. In Sect. 46.3, we introduce the system architecture and the data scheduling processes. Then, we focus on describing implementation of AGA scheduling in Sect. 46.4. In Sect. 46.5, several experiments are done to evaluate the performance of AGA scheduling strategy. Finally, Sect. 46.6 gives the conclusion and future work.

46.2 Related Works

Nowadays, a lot of data scheduling strategies have been proposed and used in many commercial P2P streaming systems. A measurement study on PPLive reports that users suffer from long startup delays and playback lags, and suggests that better segment scheduling algorithms are required [1]. Sequential scheduling [2], Random scheduling [3], and Rarest First scheduling [4] are most used, but they do not consider the demand of other peers, which need to use the segments that are received from source nodes and advertise to the network, so the network nodes can not be most utilized. The author of [5] considers the supply and demand of other peers and greatly improves the throughput of the network, but he does not consider the service performance of each peer, so this strategy can not guarantee the quality of service. The author of [6] uses the Genetic Algorithm in data scheduling and it

greatly reduces the time of scheduling, but the author assumes that each resource node has all of the data segments, so it is not suitable for the actual system. Also the algorithm easily leads to local optimum and premature convergence, so it may not find the optimal solution. The author of [7] converts the original data scheduling problem to a problem of finding a maximum match on the corresponding bipartite graph, and then assigns data packets to neighbor nodes according to the maximum match. However, the computational complexity of this method is not suitable for real-time systems.

The authors of [8] propose a weighted round-robin algorithm based on source nodes' bandwidth. In their algorithm, the number of segments assigned to each source node is proportional to its bandwidth. In other words, source nodes with higher bandwidth will be assigned more segments than those with lower ones. The authors of [9] propose to assign each segment to the source node that will deliver it the earliest. But all these algorithms do not provide any performance guarantees on user perceived visual quality and may not perform well in on-demand streaming systems [10].

46.3 Description of Data Scheduling Problem

46.3.1 System Architecture

The architecture of the P2P VOD system designed by us is shown in Fig. 46.1. We apply a tracker server and a media server. The media server contains the entire videos which are divided into segments. Each segment is of equal size and has a unique segment ID. Every peer will contact the tracker sever to join the P2P network, the tracker stores the IP address of all peers in the system and the information of super peers when they join in the system. Each super peer is always online and has a public network address. And it manages the information of peers which have the specific media resources. The normal peer which has a globally unique identifier can play the media, buffer the data segments and publish media segments.

When a new peer joins in the system, it works as follows:

- (1) The peer registers to the tracker server, and sends a message including IP address and port to tracker server. The peer also informs the tracker server of resources the peer itself caches, and then the tracker sever returns some super peers which manage these resources.
- (2) The peer notifies the cache resource information to the super peers returned by tracker server.
- (3) When one peer requests a media to play, the peer first sends the request to the tracker server to ask that which super peers manage the media resource. Then the tracker server returns a set of super peers.

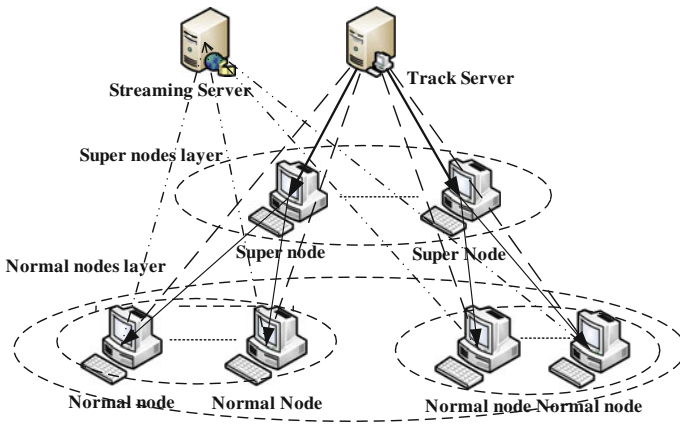


Fig. 46.1 System architecture of P2P VoD

- (4) The requesting peer queries the super peers about the media resource, and then the super peers return the peers who have the media resource, and the peers which are in the same or nearby regional location with requesting peers have priority to be return.
- (5) The requesting peer establishes a connection with each peer returned by super peers. Peers connected with the requesting peer are called neighbors. Then the requesting peer exchanges the buffer map (BM) information with its neighbors and sends request to its neighbors.
- (6) After receiving the data segments from its neighbors, the requesting peer caches and plays the media. The requesting peer informs the super peers that it has the resource segments when it receives the first data segment of this resource.

46.3.2 Evaluation Model of Source Node Selection

The aim of using scheduling algorithm is to schedule the special data segments as little time as possible and make best use of resource nodes in network. Sending time of each data segment can be described in Eq. (46.1):

$$T_{i,j} = SSize_i / EBw_{j,k} \tag{46.1}$$

where $SSize_i$ is the size of each segment. Since the network delay and the burden of each node should be considered, so we introduce a modification function to modify the time consuming of data scheduling. The modification function of scheduling cycle m is shown as follows:

$$T_m = \sum_{j=1}^n (TL_j + (S_{i,j} - 1)TW_j) \quad (46.2)$$

In the formula (46.2), n is the number of source nodes taking part in this scheduling cycle; TL_j is the delay time from source node j to requesting node. $S_{i,j}$ is the number of data segments those are scheduled in the node j in i cycle; TW_j is the waiting time of one scheduling request in queue on the peer j .

Therefore, we set $TW_{a_{ij}}^m$ as the time used to schedule the data segments in cycle m , which equals the sending time of each data segment, $T_{i,j}$, plus the modification value, T_m , the equation is expressed as follows:

$$T_{a_{ij}}^m = \sum_{i=1}^n T_{i,j} + T_m \quad (46.3)$$

The object of using scheduling algorithm is to find out the minimum $T_{a_{ij}}^m$, but the availability of the data segments in the network and the bandwidth of each peer are uncertain, so it is essentially an NP complete problem [11]. In the next section, the AGA is used to solve this problem.

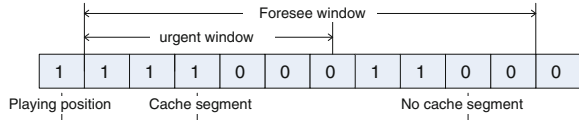
46.4 Implementation of Adaptive Genetic Algorithm

46.4.1 Segment Selection

In our P2P VoD system, videos are divided into segments with equal size. To describe the play order, every segment is assigned a number (start from zero). Each peer has all or part of the media content and maintains a BM shown in Fig. 46.2.

The BM is used to represent the available information of segments. The BM is consisted of some bits, and each bit is corresponding with a segment. If $BM[i][j] = 1$, it means that peer i has the segment j ; otherwise, peer i has no segment j . The requesting node exchanges BM and playback point with its neighbor peers, so it knows which data segments are available and needed by neighbor peers. In the strategy of our paper, an emergency window and a foreseeing window are defined in Fig. 46.2. They start from the playback point. The size of foreseeing window is larger than the size of emergency window. On the one hand, in order to guarantee the quality of service, the data segments which are not cached in emergency window must be scheduled immediately. On the other hand, in order to meet the needs of other neighbor peers and take most use of the peers in the network, the non-cached data segments behind the emergency window are selected to be scheduled according to their ratio of supply and demand. The requesting peer uses a value function to evaluate the usability value of each non-cache data segments behind the emergency window, the data segment with a higher usability value means that it is more useful for the other neighbors. A certain number of pre-fetching data segments are selected

Fig. 46.2 Information of buffer map



according to the usability value. The neighbors’ foreseeing window is defined to predict whether the data segments are needed by neighbors in the near future. We assume that a data segment will be soon scheduled by a neighbor peer if the segment is in the neighbor’s foreseeing window. The need degree of each data segment can be calculated by formula (46.4).

$$D_i = \sum_{j=1}^{n-m} C_{i,j} + \sum_{j=n-m+1}^n \lambda C_{i,j}, (0 < \lambda < 1) \tag{46.4}$$

In formula (46.4), the value $C_{i,j}$ represents how urgently the segment i is for the neighbor j . If the segment is not in neighbor’s foreseeing window, the value $C_{i,j}$ is zero. $\sum_{j=1}^{n-m} C_{i,j}$ is the sum of urgency value of data segment i needed by the neighbors which are in the same regional location with the requesting peer,

$\sum_{j=n-m+1}^n \lambda C_{i,j}$ is the sum of urgency value of data segment i needed by the neighbors

which are not in the same regional location with the requesting peer. The usability value, V_i of each segment can be calculated by requesting peer using formula (46.5). R_i is the number of replicas of data segment i among its neighbors.

$$V_i = D_i/R_i \tag{46.5}$$

Finally, we sort the data segments according to their usability value and select some segments with higher usability value to be pre-scheduled.

46.4.2 AGA Scheduling

In the previous section, we describe which data segments should be choose to be scheduled. In this section, we will introduce the AGA based on the mechanism of natural selection and genetics to determine which peer is selected to transmit the special data segment.

• A. Chromosome Encodings

The first step to apply AGA to data scheduling problem in P2P VOD system is encoding. We design an encoding for our problem as follows.

First, we sort the data segments which are scheduled in this cycle according to their index number from small to large in the queue and count the BM information

of neighbors, so we get a two-dimensional array which is not equal in length to store information and shown in formula (46.6).

$$A_{(n,s_n)} = \left\{ (a_{(1,1)}, a_{(1,2)} \dots a_{(1,s_1)}) (a_{(2,1)}, a_{(2,2)} \dots a_{(2,s_2)}), \dots (a_{(i,1)}, a_{(i,2)} \dots a_{(i,j)} \dots a_{(i,s_2)}) \dots (a_{(n,1)}, a_{(n,2)} \dots a_{(n,s_n)}) \right\} \quad (46.6)$$

Here, i denotes the serial number of data blocks in the queue; j represents the index of the owner of the i th data segment in the queue. S_i represents the total owners number of the i th data segment in the queue. $a_{i,j}$ represents the j th owner of the i th segment in the queue. For example, $a_{(3,8)} = 5$ represents that the peer with ID number 5 is the 8th owner of the 3th data segment in queue. As the scheduling order in one cycle is not important, therefore, the owner index of each data segment in the two-dimensional array is encoded, and the encoded sequence corresponds to the order of data segments in the queue. For example, if a peer wants to schedule data segments from 1 to 6, and the Chromosome encoding is $\{1, 3, 4, 3, 6, 7\}$ which represents the segment 1 is scheduled from owner $a_{(1,1)}$, the segment 2 is scheduled from owner $a_{(2,3)}$ and it is similarly to remain segments.

• B. Population Initialization

After selecting the data segments to be scheduled in one cycle, we sort them based on the sequence of each data segment. Then we select the owner index of each data segment randomly to spawn N_{pop} chromosomes by the way discussed in section A.

• C. Fitness Function and Evaluation

The AGA does not use external information during the search process, and only use the fitness value of each individual in the population. The fitness value is evaluated by a fitness function which we discussed in the Sect. 46.3. The associated fitness value is the total time that is spent on scheduling all segments in one cycle.

• D. Selection Methods

In this paper, we adopt the elite selection strategy. The idea of the strategy is to copy the best individual (called elite individual) of the group directly to the next generation during the evolutionary process.

• E. Crossover Operation

The main purpose of crossover operation in data scheduling is to generate a new scheduling coding individual by combining features of two selected scheduling coding parents. In this operation, pairs of scheduling coding individuals are selected at random from the current population and then single point crossover operation is done according to the crossover ratio. What plays a central role is the crossover ratio p_c . Here the adaptive p_c is adopted. Formula (46.7) is used to calculate the crossover ratio.

Fig. 46.3 Example of crossover operation

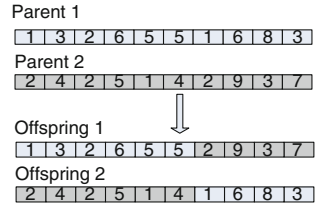
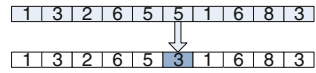


Fig. 46.4 Example of mutation operation



$$p_c = \begin{cases} 0.9 - \frac{0.3(f' - f_{avg})}{f_{max} - f_{avg}}, & f' \geq f_{avg} \\ 0.9, & f' < f_{avg} \end{cases} \quad (46.7)$$

Here f_{avg} is the average fitness value of each generation; f' is the maximum fitness value of the two individuals which will cross with each other. f_{max} is the biggest fitness value in each generation of data scheduling. A crossover operation example is shown in Fig. 46.3.

• **F. Mutation Operation**

Just like evolution in nature, chromosomes also should take a mutation operation after the crossover operation. Its main function is to ensure that the AGA has local random search capabilities, maintains the diversity of population and prevents the emergence of non-mature convergence. Here the adaptive mutation ratio p_m is used and shown as follows:

$$p_m = \begin{cases} 0.1 - \frac{0.09(f_{max} - f)}{f_{max} - f_{avg}}, & f \geq f_{avg} \\ 0.1, & f < f_{avg} \end{cases} \quad (46.8)$$

Here f represents the value of individual fitness. A mutation operation is shown in Fig. 46.4.

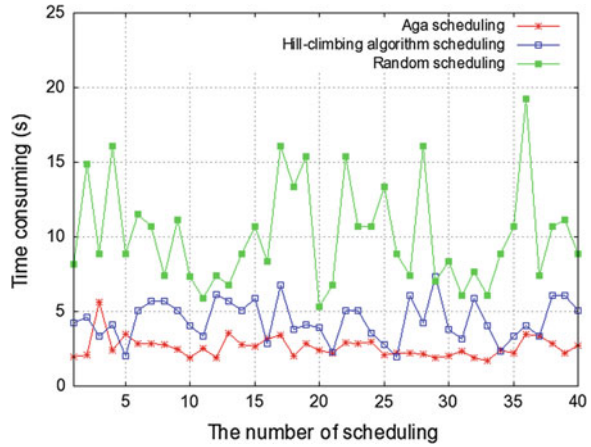
• **G. Stopping Condition**

The evolution stops if the total number of iterations reaches a predefined number of iterations [12]. In this paper, we define the number 50.

46.5 Simulation and Evaluation

We use simulations to evaluate the performance of scheduling strategy. We assume that each resource node has a certain number of neighbors, and each neighbor node has some segments of data. The requesting peer starts to watch

Fig. 46.5 Time consuming of data scheduling



video from the beginning. We set the delay time from the neighbor nodes to the requesting node ranging from 30 to 100 ms randomly. The uplink bandwidth of neighbor nodes allocated to the requesting node is randomly set between 30 and 1,000 Kbps. The waiting time of processing the next request is randomly set between 50 and 100 ms. The simulation does not consider the segment loss. The media file is divided into segments, and the size of each segment is 32 KB. We assume that the total number of segments of the media resource is 2,000, we schedule 50 segments data in each scheduling cycle, so the total cycles of scheduling is 40.

In the experiment of the time consuming of data scheduling, the scheduling strategy based on AGA not only considers the priority of the data segments to be selected in one scheduling cycle, but also allows for the factors such as capacity and bandwidth of the neighbor nodes. In the experiment, we respectively use three kinds of scheduling strategy which are AGA, hill-climbing algorithm scheduling and random scheduling algorithm to generate three scheduling time sets. The results are shown as Fig. 46.5. The vertical axis represents the number of scheduling cycle, the vertical axis represents the time consuming of each scheduling cycle.

From the Fig. 46.5, we can see that the time consuming of data scheduling of random scheduling algorithm is longer and has a wide fluctuation in scale, while the time consuming of data scheduling of scheduling strategy based on AGA is significantly shorter and scheduling time has a very small fluctuation. The hill-climbing algorithm scheduling is better than random scheduling, but it is a local search algorithm, so the result is not better than the AGA.

In addition, we simulate the utilization of the uplink bandwidth. The utilization of the uplink bandwidth is the total bandwidth that the source nodes assign to the requesting node to transmit data content. The utilization of uplink bandwidth is continuously calculated with the number of neighbor nodes increasing. The result is shown in Fig. 46.6, Where the horizontal axis represents the number of neighbor

Fig. 46.6 Utilization of uplink bandwidth

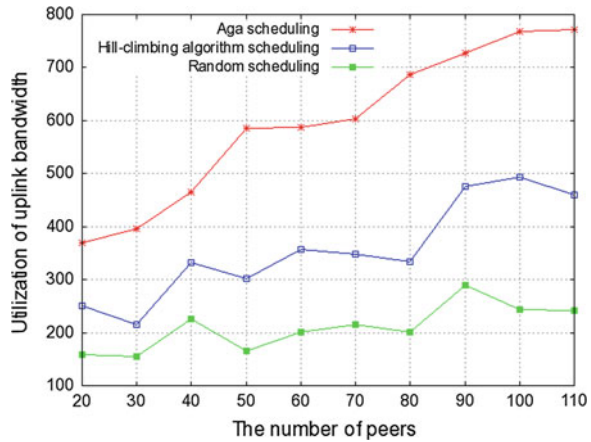
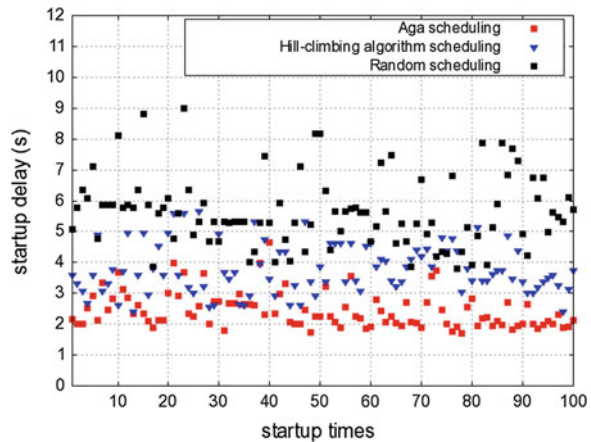


Fig. 46.7 Comparison of startup delay



nodes, the vertical axis indicates the size of uplink bandwidth that utilized by requesting peer.

Figure 46.6 shows that the actual uplink bandwidth utilization raise greatly with the number of neighbor nodes increasing in the scheduling strategy based on AGA, so the AGA scheduling can quickly increase the number of the data segment backup in network and improve the performance of system. While the actual utilization of uplink bandwidth increase slowly with the number of neighbor nodes increasing in random scheduling strategy and hill-climbing algorithm scheduling strategy.

Figure 46.7 shows the effect of three scheduling algorithms on the startup delay. Here the startup delay is defined as duration of requesting peer accepting first 30 segments. We all took startup operations for 100 times under three data scheduling strategy respectively and recorded the delay time every time. The

horizontal axis represents the startup times, the vertical axis indicates the startup delay.

From the Fig. 46.7, we can see that the startup delay of random scheduling algorithm is longer and has a wide fluctuation in scale, while the startup delay of scheduling strategy based on AGA floats up and down in value 2.5 s. It is the best of three scheduling algorithms. Since the hill-climbing algorithm scheduling is a local search algorithm, so the startup delay is longer than the AGA.

46.6 Conclusions and Future Work

In this paper, a scheduling strategy based on AGA is proposed. The scheduling strategy not only considers the segments' urgency and the ratio of segments' supply and demand, but also takes account of the factor of regional location to select the segments which is needed to schedule in a scheduling cycle. After the scheduling strategy selected the needed segments to schedule, it utilizes the AGA to select the suitable source nodes to transmit the selected segments. Finally, the simulation experiments are done to show that our scheduling strategy can reduce the time consuming of each data scheduling and startup delay, make the most use of the network uplink bandwidth, increase the number of data segments backup in the network rapidly and improve the service performance of the system greatly.

In our future work, we will study more details about AGA scheduling strategy in actual systems and how to retransmit the data segments in the case of segments loss.

Acknowledgements This work was supported by the National Natural Science Foundation of China (No. 61063038), the Foundation of Key Laboratory of Guangxi Trusted Software (No. kx201114), and the Foundation of Key Laboratory of Guangxi Wireless Broadband Communication and Signal Processing (No. 11108).

References

1. Hei X, liang C, liang J, Liu Y, Ross K (2007) A measurement study of a large-scale P2P IPTV system. *IEEE Trans Multimed* 9(3):405–414
2. Zhang X, Liu J, Li B, Yum TSP (2005) CoolStreaming/DONet: a data-driven overlay network for live media streaming. *IEEE INFO COM*, Miami
3. Venkataraman V, Yoshida K (2006) Chunkyspread: heterogeneous unstructured tree-based peer-to-peer multicast. *IEEE International Conference on Network Protocols(ICNP)*
4. Locher T (2007) Push-to-pull peer-to-peer live streaming. *21st international symposium on distributed computing (DISC)*. pp 388–402
5. Fang S-C (2008) A supply-and-demand based scheme for peer-to-Peer video-on-demand system
6. Huang G, Li C, Zhou Y, Bin C (2011) Data scheduling strategy in P2P VoD system based on genetic algorithm. *2011 third international conference on multimedia information networking and security*

7. Wu J, Peng Y, Liu F (2011) Transmission scheduling in data-driven peer-to-peer streaming towards optimal throughput. 2011 IEEE international conference on networking, architecture, and storage
8. Agarwal V, Rejaie R (2005) Adaptive multi-source streaming in heterogeneous peer-to-peer networks. SPIE/ACM multimedia computing and networking, San Jose, pp 13–15
9. Kowalski G, Hefeeda M (2009) Empirical analysis of multi-sender segment transmission algorithms in peer-to-peer streaming. IEEE international symposium multimedia (ISM'09), San Diego, CA, pp 243–250
10. Annapureddy S, Guha S, Gkantsidis C, Rodriguez P (2007) Is high-quality VoD feasible using P2P swarming? World Wide Web Conference (www'07), Banff, AB, Canada, pp 903–912
11. Brucker P, Jurisch B, Sievers B (1994) A branch and bound algorithm for job-shop scheduling problem. *DiscrAppl Math* 49:105–127
12. Srinivas M, Patnaik LM (1994) Genetic algorithms: a survey. *IEEE Comput* 27:617–626