Turkish Journal of Electrical Engineering & Computer Sciences

http://journals.tubitak.gov.tr/elektrik/

Research Article

Turk J Elec Eng & Comp Sci
(2017) 25: 877 – 887

doi:10.3906/elk-1506-2

A new mechanism to improve video streaming in P2P networks using helper nodes

Javid KARGAR[1], **Abdorasoul GHASEMI**[2,*]
[1]Faculty of Computer and Information Technology Engineering, Qazvin Branch, Islamic Azad University, Qazvin, Iran
[2]Faculty of Computer Engineering, K. N. Toosi University of Technology, Tehran, Iran

Received: 01.06.2015 • **Accepted/Published Online:** 03.03.2016 • **Final Version:** 10.04.2017

Abstract: Peer-to-peer video on demand (P2P-VoD) is a solution to provide video service to a large number of users on the Internet. Due to the upload bandwidth bottleneck of VoD servers, the streaming capacity in the P2P-VoD system is limited. This paper proposes a new mechanism to improve the video streaming in P2P networks by using the idle peers upload bandwidth and storage, which are called helpers. The proposed method uses an XOR mechanism in the VoD server to encode chunks and reduce bandwidth use. The helpers receive encoded chunks from the VoD server and save them in their storage spaces and cooperate in sharing these chunks with the other peers. Simulation results using OMNET++ are provided to show the efficiency of the proposed mechanism in terms of the required upload capacity requirement of the server and deploying the storage and bandwidth of helpers. The results of simulation show that the upload bandwidth usage of the server has been reduced. On the other hand, the proposed method increases the memory usage of helpers.

Key words: Video streaming, peer-to-peer networks, video on demand, bandwidth usage

1. Introduction

In recent years, a high-speed Internet connection with high bandwidth has been provided for users in both developed countries and even in developing countries. Nowadays most users want to watch video on the Internet. One challenge is supporting such a large and diverse on-demand request and reducing its cost [1]. It is clear that architectures towards centralized distribution and centralized control are not likely to be suitable as video adoption grows. With the increasing spread of the Internet around the world it has been proved that a peer-to-peer (P2P) network is a good technology to share and transfer files and stream videos [2]. In P2P solutions, the required server upload bandwidth deceases and the centralized storage and scalable solutions are achievable using distributed solutions [3]. The P2P system also can be used in video on demand (VoD) services. In a VoD system, peers are able to pause the video or even perform some random seeks to an arbitrary point of playback in the video [4]. There are some P2P systems for content delivery such as BitTorrent [5], PPLive [6], UUSee [7], and YouTube [8]. Because of increasing video quality, peers need to have upload bandwidth capacity to satisfy the demand alone [3]. Therefore, designing a new approach is essential to improve the performance of these technologies. In P2P networks, there are many online users with spare upload bandwidth, which are called helpers, and can be used in the P2P-VoD system. They work with users in a P2P network to deliver

*Correspondence: arghasemi@kntu.ac.ir

the content. Helpers would not change the chunk content in the P2P-VoD systems and they just improve the performance of P2P-VoD systems [3].

In P2P-VoD systems, the server or helpers can use network coding. It combines segments and sets a schedule for distributing chunks in network, and, as a result, the system will achieve a better performance compared to a traditional P2P system [9]. In addition, network coding reduces the bandwidth use of servers in P2P systems [10].

This study addresses the VoD problem using the P2P architecture. The main contributions of this paper are as follows: developing a new algorithm with network coding and a helper caching mechanism for reducing workload and bandwidth use of VoD servers in P2P-VoD systems and achieving the maximum bandwidth amplifying effect of helpers by using a targeted communication between nodes.

The organization of this paper is as follows. Section 2 reviews the related works. Section 3 first discusses about basic model and introduces helpers into P2P-VoD, and then proposes the new method. Section 4 presents the simulation results and section 5 concludes the paper.

2. Related work

This section demonstrates some related works that use either helper or network coding in P2P-VoD systems. One of the first works is introduced in [11], in which the authors use idle peers that are interested in content delivery. In this scheme, the helpers can enhance the capacity of the P2P network. However, the helper may download a large portion of the file and this leads to resource wastage in terms of storage and bandwidth. In [12], the authors design a system that is called TRIBLER; it uses helpers based on social relations. In fact, it is a social-based peer-assisted P2P system. The authors design a novel paradigm for P2P file-sharing networks based on trust or friendship. One of the advantages of this method is that it is designed and implemented in accordance with the BitTorrent system. In [13], the authors discuss the mechanism of TRIBLER through a practical system, which is called 2Fast. The focus is on improving the data transfer rate. The authors suppose that the data on the network are in the form of files or archive files. In 2Fast, any peers have one of two roles, either a collector or a helper. One of the advantages of 2Fast is that it eliminates the bottleneck of the upload bandwidth capacity, because it can limit the download speed of peers. In [14] the authors design a BitTorrent-like file download system using helpers. The authors analyze the performance of system with helpers. In fact, they propose a content distribution mechanism with helpers in which each helper just can download a tiny fraction of a file. Their method presents a low storage cost to the helpers and attempts to implement helpers to improve the file downloading systems. However, if peers require a large file, they have to download very small pieces of a file several times and it increases overheads. In [7] the authors present a new protocol with network coding for segment scheduling in the UUSee for P2P-VoD systems. In terms of reducing server bandwidth usage it proves that network coding combined with segment scheduling strategy can improve performance. In [15], each peer that wants to receive a segment sends a request to the other peers in the neighbor list or to the helper. Then the helper gathers the requests and selects a subset D of requests that consists of m segments. The helper sends a request to the server and the server sends the corresponding XOR encoded segment of these m segments to the helper. The helper relays this encoded segment to the chosen subset. In this scheme, each peer that receives the encoded segment must have the other $m-1$ segments in subset D to successfully decode the missed chunk. In addition, the computational load of the helper increases, because it has to select a proper subset of requests with respect to the cache information of peers. On the other hand, in this method, helpers just relay the encoded segment and do not use a cache mechanism to avoid sending requests to the server repeatedly.

In [16], the authors use distributed algorithms for helper nodes to connect to a set of interested peers in order to balance their storage and bandwidth resources. The helpers use a random linear combination for coding each packet that they store for each segment. Each peer runs the algorithms independently, which increases the computational load of the peers.

In [3] the authors develop a new theoretical model to study the performance of large-scale P2P-VoD systems in terms of server loading.

This paper combines network coding and a helper-caching mechanism in a P2P-VoD system to decrease bandwidth use of a VoD server and peers. In addition, this paper introduces a targeted communication between peers, helpers, and VoD server to increase the use of helpers' capacities.

3. System description

This section presents the basic idea and the proposed model for decreasing average server loading of P2P-VoD systems by helpers.

3.1. Basic model

This section demonstrates the basic model for large-scale P2P-VoD systems. Figure 1 depicts the video streaming model with helpers. Peers who are viewing the video will assist in re-distributing the content to other peers. It is assumed that there are N_p peers in the system that contribute in video streaming. In addition, a swarm in a P2P-VoD system with a VoD-server, tracker, helpers, and a large number of peers is considered. The video is divided into m segments. As shown in Figure 1, helpers are placed between peers and VoD server. If peers receive a request from other peers and if they have the requested chunk in their storage, they will send it to them. If peers send a request to helpers and if the helpers have the requested chunk, they will send it to the peers. Otherwise, they send a request to the server, get the chunk from the server, and then relay it to the peers. As shown in Figure 2, the VoD system also uses a tracker. The tracker assists in communication between nodes and keeps track of where file copies reside on node machines, which ones are available at time of the client request, and helps coordinate efficient transmission. The tracker also contains the information of all peers and their location, manages the system, and makes the overlay.

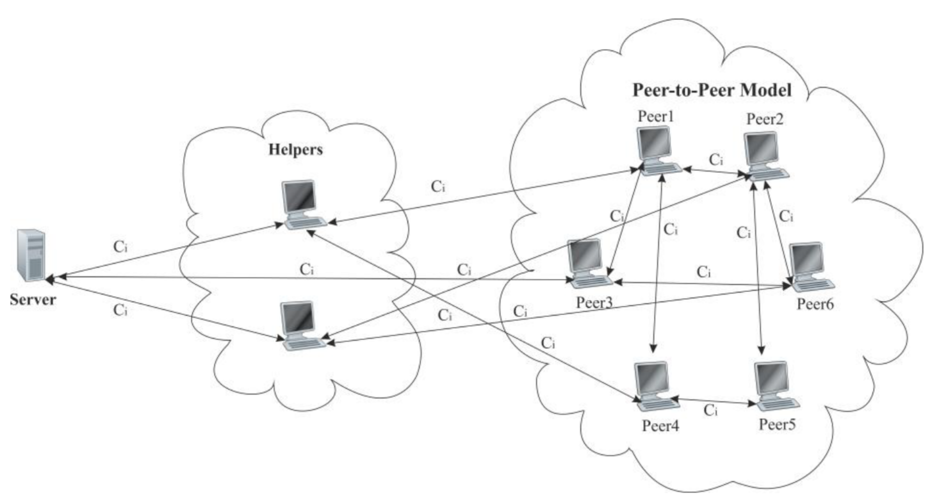

Figure 1. Video streaming with helpers.

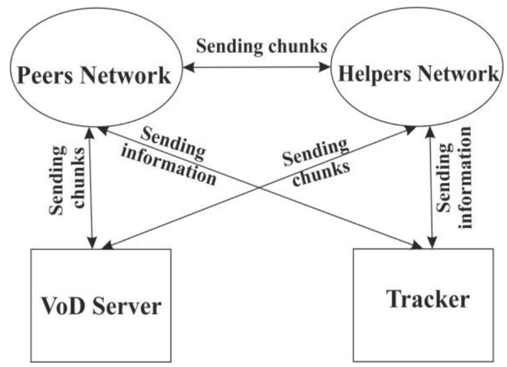

Figure 2. P2P-VoD system architecture.

3.2. Proposed model

This section presents the proposed model based on network coding for P2P-VoD systems and proposes a new content delivery mechanism as depicted in Figure 3. The proposed model uses network coding for sending chunks from the VoD server to helpers and from helpers to peers. When a user joins the system, it obtains the helpers list from the tracker. The peers also maintain connections to a helper to enhance the streaming rate and get the chunks that cannot be obtained by their neighbor's peers. As shown in Figure 3, in the proposed P2P-VoD system, there are n chunks of the same size and a group of peers that initiate delivery request for helpers or the server. Peers in some cases send a request only to helpers and receive an encoded chunk rather than a simple chunk. Actually, the motivation of the proposed scheme is based on this observation that in the current schemes that are using helper nodes in many cases there are many duplicated chunks in the helper nodes. Furthermore, it is probable that a peer has the previous and the next chunk of a needed chunk. Therefore, it would be beneficial that the helpers ask for coded chunks of the VoD server and relay this chunk to the peers. The improvement in efficiency depends on the probability that a peer has the previous and the next chunk of the needed chunk and the probability that the corresponding helper has the encoded chunk. In this scheme as helpers and peers receive more chunks, these probabilities increase. Therefore, the number of peers' requests toward the server is reduced. As a result, using XOR base network coding and helper base P2P-VoD streaming can decrease the average upload bandwidth use of the VoD server and hence reduces the average time that peers have to wait to receive chunks.

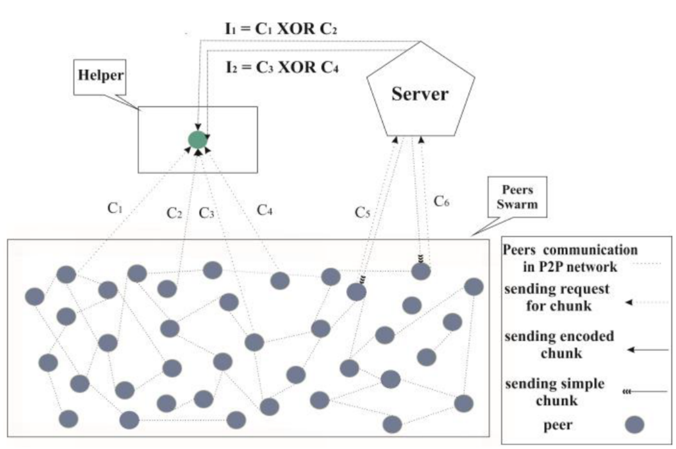

Figure 3. The proposed model.

The proposed method is explained with an example of the memory cache of peers and helpers. As shown in Figure 4, peers 15, 20, and 25 have the desired condition in the proposed method. Peers 15 and 20 have chunks 12 and 14 but they do not have chunk 13. In addition, peer 25 has chunks 13 and 15 but it does not have chunk 14. In that condition, peers 15, 20, and 25 send a request for chunks 13 and 14 to a helper and the helper already had the encoded chunk obtained from the VoD server; thus the helper can send the encoded chunks to the peers. In another condition, peers 30 and 35 send a request to a helper to receive chunks 15 and 16, respectively. However, the helper does not have them in storage; thus the helper sends a request for the VoD server to receive the encoded chunk. The server compresses these two chunks into one chunk with the XOR method and sends it to the helper, and then the helper stores it and sends it to the peers. The peers who receive the encoded chunk must have the other chunk to decode it. Since each chunk is encoded with the next or previous chunk in the server and each peer in the desired case has the previous and next chunk that they demand for it, they could simply decode every encoded chunk and retrieve their chunk. The proposed method can reduce the upload bandwidth use of the server. Because the server compresses two chunks into one chunk and sends one chunk instead of two chunks in some cases and the helpers can store either encoded or simple chunks and if other peers have the desired condition they send a request to helpers, they can receive an encoded chunk from helpers. The pseudo code of the proposed method is shown in Figure 5.

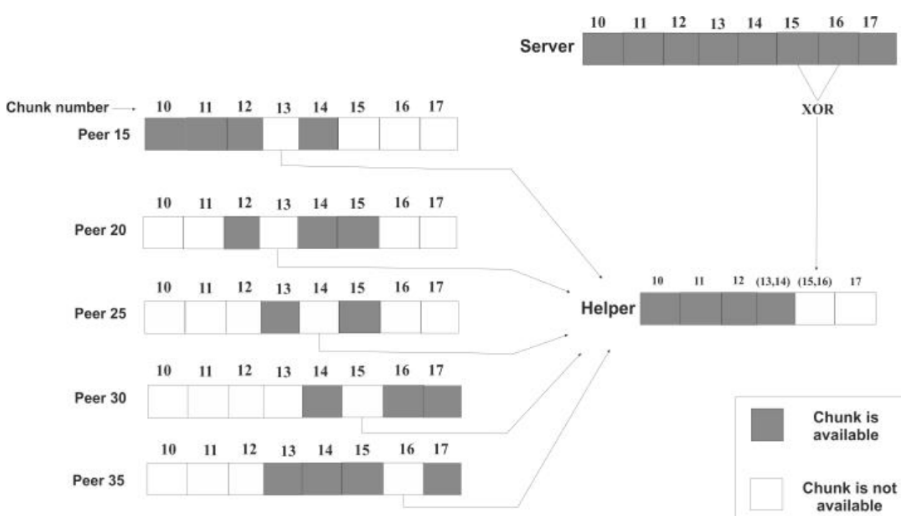

Figure 4. Example of peer cache.

This paper assumes that there are C_i i = 1... n video chunks transferred between the VoD server and other nodes. In the scheme without using an encoding method, the server has to send all of the requested chunks to other nodes and it has to consume most of its upload bandwidth. Hence, we can compute the upload bandwidth use of the VoD server in this case as follows:

$$\mu_s = \sum_{i=1}^{N_{dr}} C_i \times \min(r, u_s) \tag{1}$$

Here N_{dr} is the number of download request to the VoD server, r is the streaming rate, and U_s is the upload bandwidth capacity of the VoD server. That is, in the P2P-VoD system, sending a chunk from the server to other nodes depends on the upload bandwidth capacity of the server and the streaming rate. Therefore, the server can send a chunk based on the minimum of these two factors.

```
If (P_i has C_i-1 and C_i+1) then // If peer_i wants chunk_i and it has chunk_i-1 and chunk_i+1
    P_i sends req to H_i; // Peer_i sends a request to helper_i
    If (H_i has I_i or C_i) then // If helper_i has the encoded chunk (I_i) or simple chunk (C_i)
        H_i sends I_i or C_i; // Helper_i sends the encoded chunk or simple chunk to peer_i
        Else   H_i sends req to Server;
        Server sends I_i; //Server sends the encoded chunk (I_i = C_i XOR C_i+1) to helper_i
        H_i sends I_i; // Helper_i sends the encoded chunk to peer_i
    End if
    Else   P_i sends req to P_j or H_i or Server;
    If (P_i sends req to P_j) then
        If (P_j has C_i) then
            P_j sends C_i; // Peer_j sends chunk_i to peer_i
            Else   P_i sends req to H_i;
            If (H_i has C_i) then
                H_i sends C_i; // Helper_i sends chunk_i to peer_i
                Else   H_i sends req to Server;
                Server sends C_i; // Server sends chunk_i to helper_i
                H_i sends C_i; // Helper_i sends chunk_i to peer_i
            End if
        End if
    End if
    If (P_i sends req to H_i) then
        H_i sends C_i; // Helper_i sends chunk_i to peer_i
    End if
    If (P_i sends req to Server) then
        Server send C_i; // Server sends chunk_i to peer_i
    End if
End if
```

Figure 5. Pseudo code of the proposed method.

For the scenario in which we use the encoding technique, the upload bandwidth use of the VoD server can be computed as follows:

$$\mu_{en} = (P_r \sum_{i=1}^{N_{dr}} \frac{C_i}{2} + (1 - P_r) \sum_{i=1}^{N_{dr}} C_i) \times \min(r, u_s), \tag{2}$$

where P_r is the probability that the helper sends a request to the VoD server for receiving encoded chunks. In this scheme, since two chunks are encoded in one transmission, the number of video chunks that the VoD server sends to the helper will be $\frac{C_i}{2}$.

Therefore, the efficiency of the proposed scheme depends on the probability P_r. When the number of chunks in the cache of a peer goes up, the probability that each peer has the desired condition of using the encoded chunks increases. For the case in which P_r is 0.5, the upload bandwidth use of the VoD server using the encoding method would be as follows:

$$\mu_{en} = \frac{1}{2}(\sum_{i=1}^{N_{dr}} \frac{3C_i}{2}) \times \min(r, u_s) \tag{3}$$

If we consider that the number of chunks (n) is 100, the upload bandwidth use of the VoD server without and with using the encoding technique are, respectively, as follows:

$$\mu_s = 100 \times \min(r, u_s) \tag{4}$$

$$\mu_{en} = 75 \times \min(r, u_s) \tag{5}$$

Therefore, as it has been shown in section 4, the upload bandwidth use of the VoD server using the proposed method will decrease to 25 $\pm$ 5% approximately.

The proposed method also reduces the average time that the peers need to wait to obtain a chunk. In this study, T_p is the average time that peers need to wait to receive m chunks, and the time that the helper delivers a chunk to the peer is t_i. The time that the server delivers a chunk to the helper is t_h. The number of chunks is m and the number of peers is N_p. Eq. (6) is for the case in which the server does not use network coding and sends a simple chunk individually to a helper.

$$Tp_1 = \frac{m \times t_h + \sum_{i=1}^{N_p} t_i}{m} \tag{6}$$

Eq. (7) is for the case in which the server sends encoded and compressed chunks to a helper, and the helper relays them to the peers.

$$Tp_2 = \frac{(\frac{m}{2}) \times t_h + \sum_{i=1}^{N_p} t_i}{m} \tag{7}$$

Eq. (8) is for the case in which a helper has already obtained the chunks from the VoD server and relays them to the peers.

$$Tp_3 = \frac{m \times \sum_{i=1}^{N_p} t_i}{m} = \sum_{i=1}^{N_p} t_i \tag{8}$$

Obviously, because $Tp_3 << Tp_2 << Tp_1$, the proposed coding algorithm can reduce the average time that peers have to wait to receive chunks. In general, the differences between the proposed method and other methods include the following:

1. Storing the compressed chunk in the storage space of helpers: In the proposed method, helpers store either a simple chunk or an encoded chunk in their storage space and if they receive duplicate requests for the same chunk that they have already received from the VoD server, they relay it to the peers and will not send a request to the server again.

2. Distributing information among helpers: In the proposed method, helpers can just receive chunks from the VoD server and they do not have any connection to each other. Because if they receive chunks from each other they will have duplicate chunks. Therefore, the proposed method sets the helpers to have a collection of distributed chunks rather than duplicate chunks.

3. Initializing an optimal connection between helpers and peers and the VoD server: In the proposed method, for some conditions that are mentioned in section 3.2, peers will receive compressed and encoded chunks only from helpers.

4. Simulation experiments

This section presents the simulation results obtained with the OMNet++ simulator and experiments to validate the proposed model developed in the previous section. A video is divided into chunks of equal length each consisting of some equal packets of the same size. This study focuses on the performance measures that include average upload bandwidth use of peers, average upload and download bandwidth use of helpers, average bandwidth use of the VoD server, and average storage use of helpers.

The maximum number of peers is set to 100. In addition, a P2P Ethernet network with UDP connections has been used and all peers have the same dual core CPUs. The numbers of helpers and peers are varying in different simulation scenarios. The values of the parameters are shown in the Table. Figure 6 depicts the simulation results for measuring upload bandwidth use of peers without helpers, in which there is no helper, without coding (base model) and with coding (proposed model). System parameters are set the same for each case. As shown in Figure 6, the upload bandwidth use of peers in both with coding and without coding cases is the same, but both of them are much less than the case in which the system does not use helpers. It happens because the upload bandwidth use of peers in the presence of helpers is low and they receive many chunks from helpers. In the absence of helpers, the average of upload bandwidth use of peers is 42%, but in the presence of helpers it is about 28%. Experiment 2 analyzes the upload and download bandwidth use of helpers. It calculates the average value of bandwidth use of all helpers. As shown in Figure 7, the highest amount of download bandwidth use of helpers in the without helper case is about 70% and the average is about 39%. However, the average of download bandwidth use is about 27%. Because the proposed method uses network coding and in some cases helpers receive encoded and compressed chunks instead of simple chunks and they can use their storage for storing chunks and they do not download duplicate chunks from the VoD server. The upload bandwidth use of helpers is shown in Figure 8. In the proposed method it is about 90% but without using coding it is about 87%, which means helpers participate more in video streaming for sending chunks to peers who need either compressed chunks or simple chunks.

Experiment 3 analyzes the average bandwidth usage of the server. In the P2P-VoD system, peers send requests to the server for receiving chunks that they cannot get from neighbors. In this experiment, the data of the server are recorded and the system performance is analyzed in two cases: using a coding algorithm and not using a coding algorithm. Figure 9 demonstrates that the upload bandwidth use on the dedicated VoD server is reduced from 75% to 64%, which means in the proposed method the server consumes less upload bandwidth. The reason is that in some cases the VoD server sends an encoded chunk instead of sending two chunks separately. Each encoded chunk consists of two unencoded chunks. In this aspect, P2P-VoD system performance is improved by 11% due to the reduction of chunks, which are delivered from the VoD server to helpers by using network coding in the system.

Experiment 4 analyzes the average storage use of helpers in the without coding and with coding cases.

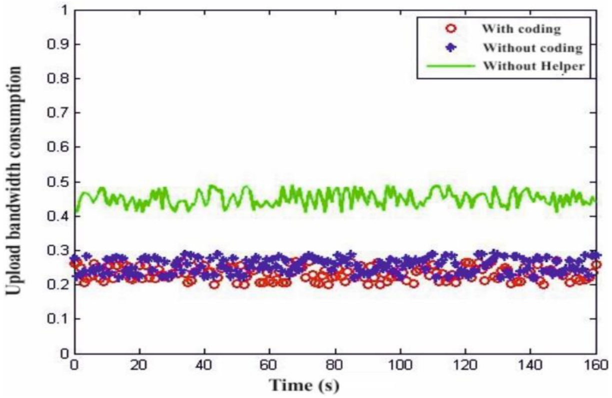

Figure 6. Average upload bandwidth usage of peers.

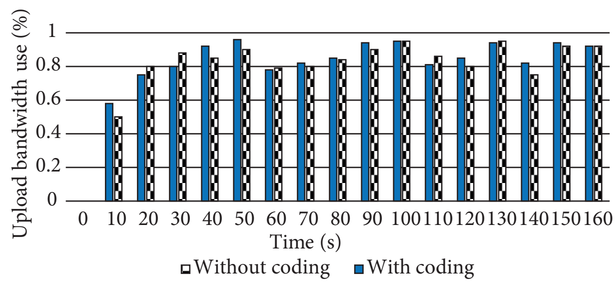

Figure 7. Download bandwidth use of helpers.

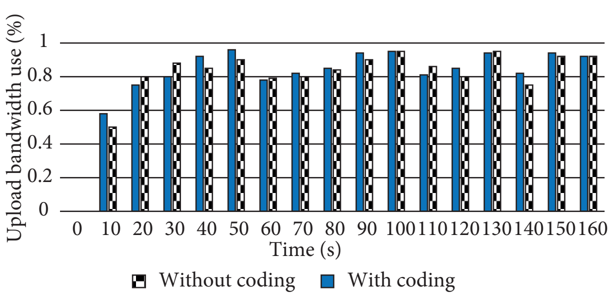

Figure 8. Upload bandwidth use of helpers.

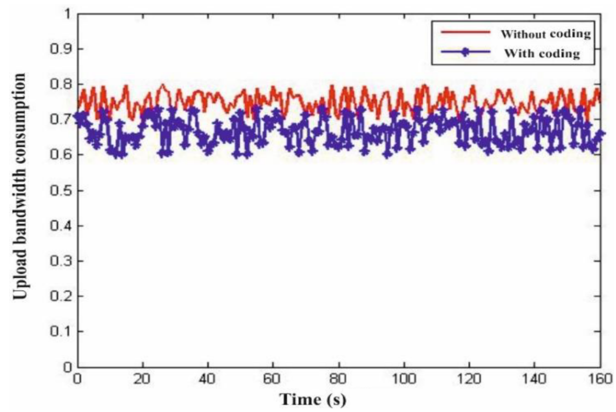

Figure 9. Upload bandwidth use of VoD server.

Table. The value of the parameters in the simulations.

N_p^{max} (the maximum number of peers)	100
N_h^{max} (the maximum number of helpers)	5
r (the streaming rate)	512 kbps
P_u (the average upload bandwidth of peers)	128 kbps
H_u (helpers upload bandwidth)	256 kbps
N_v (the number of video files)	2
S_v (the size of each video file)	4 MB
C_h (the cache size of helpers)	8 MB
C_p (the cache size of peers)	8 MB
V_l (Each video length)	250 s
L_{ch} (the length of each chunk)	5 s
N_n (the number of neighbors)	$3 \leq N_n \leq 6$
N_{dr} (the number of download request for each node)	$1 \leq N_{dr} \leq N_n$
P_t (the playback time of each peer)	$0 \leq P_t \leq V_l$

As shown in Figure 10, the storage use of helpers has been increased 10% compared to the without coding case. The reason is that in the proposed scheme helpers have to store both simple chunks and encoded chunks. For example, when some peers send requests to a helper for the same chunk at the same time but with different conditions and the helper does not have the requested chunk in storage, the helper has to receive both encoded and simple chunks from the VoD server, store them, and relay them to the peers. Therefore, in that condition, helpers have to use their storage more and the average storage use of helpers will increase. Experiment 5 analyzes the upload bandwidth use of the VoD server with different numbers of peers and helpers. As shown in Figure 11, if the maximum number of peers increases in the system, they will consume more upload bandwidth of the VoD server. The upload bandwidth use of the VoD server with different numbers of helpers is shown in Figure 12. As the number of helpers is increased, VoD server workload and its bandwidth use will decrease. In this experiment, the maximum number of peers (N_p) is set to 100. The simulation results show that if the number of helpers is set to five the proposed method can achieve maximum amplifying effect of helpers.

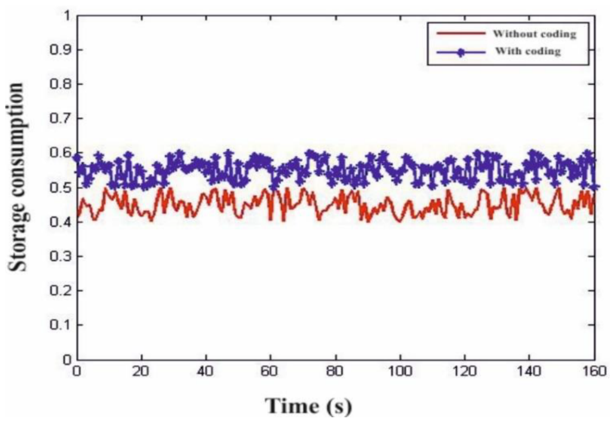

Figure 10. Storage use of helper.

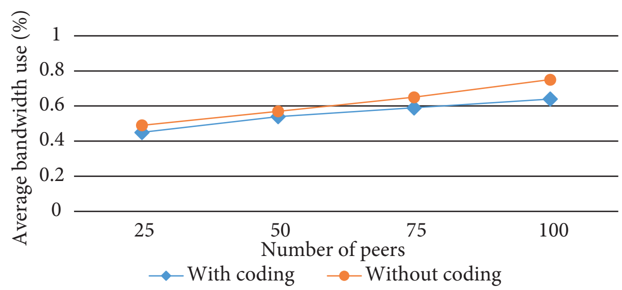

Figure 11. Upload bandwidth use of server with different numbers of peers.

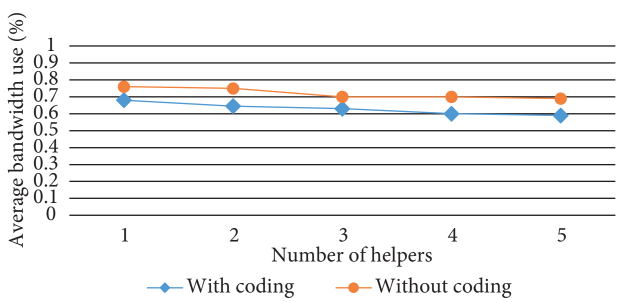

Figure 12. Upload bandwidth use of server with different numbers of helpers.

In the proposed method, peers more frequently obtain their requested chunks from helpers instead of the VoD server. This causes the experienced average latency of peers to become less. In the proposed method, when peers receive an encoded chunk they have to decode it using its next or previous chunk. Therefore, peers should consume more computing resources to find the needed chunk from the received encoded one. In addition, according to Figure 10, helpers have to use their storage space to store both unencoded and encoded chunks.

5. Conclusion

We present a new coding algorithm for P2P-VoD. The proposed mechanism reduces the upload bandwidth use on the dedicated VoD server by implementing network coding. However, it increases the required storage use of helpers because in some conditions helpers receive both unencoded and encoded chunks from the VoD server. The proposed method uses 90% of helpers' upload capacity and enhances the quality of the VoD system. It decreases the average time that each peer has to wait to receive all chunks and because in the proposed method there is no connection between helpers the number of duplicated chunks will decrease. In the future works we will design an incentive mechanism for peers with idle and high upload bandwidth to join the helpers group.

References

[1] Huang Y, Fu T, Chiu D, Lui M, Huang J. Challenges, design and analysis of a large- scale P2P-VoD system. In: Proceedings of the ACM SIGCOMM 2008 Conference on Data Communication; 17–22 October 2008; New York, NY, USA. pp. 375-388.

[2] Gummadi K, Dunn R, Saroiu S, Gribble S, Levy H, Zahorjan J. Measurement, modeling, and analysis of a peer-to-peer file-sharing workload. In: Proceedings of the 19th ACM symposium on operating systems principles; 19–22 December 2003; New York, NY, USA. pp. 314-329.

[3] Zhang Y, Wang H, Li P, Jiang Z. Modeling analysis of server loading in large-scale P2P-VoD systems with actively collaborative helpers. Int J Commun Sys 2013; 54: 215-227.

[4] Huang C, Li I, Ross KW. Can Internet video-on-demand be profitable? In: Proceedings of the 2007 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications; 27–31 August 2007; New York, NY, USA. pp. 133-144.

[5] Giannetti I, Anastasi G, Conti M. Energy-efficient P2P file sharing for residential BitTorrent users. In: 2012 IEEE Symposium on Computers and Communications; 1–4 July 2012; Cappadocia, Turkey: IEEE. pp. 524-529.

[6] Qing Y, Changjia C. A study on topology model and data contribution strategy of PPLive. In: International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery; 10–12 October 2010; Huangshan, China: IEEE. pp. 301-304.

[7] Liu Z, Wu C, Li B, Zhao S. UUSee: large-scale operational on-demand streaming with random network coding. In: INFOCOM 2010 proceedings; 14–19 March 2010; San Diego, CA, USA: IEEE. pp. 1-9.

[8] Cheng X, Mehrdad F, Ma X, Liu J. Understanding the YouTube partners and their data: measurement and analysis. China Commun 2014; 11: 26-34.

[9] Zeng H, Huang J, Tao S, Cheng W. A simulation study on network coding parameters in P2P content distribution system. In: Third International Conference on Communications and Networking; 25–27 August 2008; Hangzhou, China, pp. 197-201.

[10] Wang H, Zhang Y, Li P, Jiang Z. The benefits of network coding in distributed caching in large-scale P2P-VoD systems. In: IEEE 2010 Global Telecommunications Conference; 6–10 December 2010; Miami, FL, USA: IEEE. pp. 1-6.

[11] Wang J. Enhancing collaborative content delivery with helpers. MSc, University of British Columbia, Vancouver, Canada, 2004.

[12] Garbacki P, Wang J, Bakker A, Pouwelse J. TRIBLER: a social-based peer-to-peer system. Concurr Comp-Pract E 2006; 20: 127-138.

[13] Iosup A, Eperma D, van Steen M, Garbacki P. 2Fast: Collaborative downloads in p2p networks. In: Sixth IEEE International Conference on P2P Computing; 6–8 September 2006; Cambridge, UK: IEEE. pp. 23-30.

[14] Yeo C, Prabhakaran V, Ramchandran K, Wang J. On the role of helpers in peer-to-peer file download systems: design, analysis and simulation. In: The Sixth International Workshop on Peer-to-Peer Systems; 26–27 February 2007; Bellevue, WA, USA, pp. 1-6.

[15] Xia L. An algorithm of using helpers to deliver content segments in P2P VoD system. In: 8th International Conference on Computing and Networking Technology (ICCNT); 27–29 August 2012; Gueongju, South Korea: IEEE. pp. 86-90.

[16] Zhang H, Chen M, Parekh A, Ramchandran K. An Adaptive Multi-channel P2P Video-on-Demand System using Plug-and-Play Helpers. In: UC Berkeley Technical Report, 2010.