# Cloud Assisted Live Video Streaming over DHT Overlay Network

Pheng Un Lim and Hwangkyu Choi
Department of Computer Science and Engineering
Kangwon National University
Chuncheon, Korea
e-mail: {limphengun, hkchoi}@kangwon.ac.kr

***Abstract*—Many researches have been proposed to solve the scalability, the availability and the low-latency problems of peer-to-peer live video streaming; however, the problems are still not yet completely solved. In this paper, we propose a cloud assisted live video streaming over the distributed hash table (DHT) overlay network to tackle these problems. Our system is based on a scalable DHT network, which structures into a mesh-based overlay, to efficiently share the video stream. Moreover, cloud assist is introduced in order to maintain the availability, the low-latency and the scalability of the system. In this work, cloud server is also part of the DHT network. The role of the cloud server is to assist other nodes when the number of nodes which received video segment is less than the specific threshold. The cloud servers directly receive video segments from video server so the quality and the availability of the video segments can be guaranteed. The evaluation results show that with the help of cloud assist, the availability and the low-latency of the video segments are greatly improved if cloud has enough bandwidth.**

***Keywords-video live streaming; cloud computing; peer-to-peer; DHT overlay network.***

## I. INTRODUCTION

Because of scalability, availability and low-latency problems, the current live video streaming still cannot meet the user demand. Many studies have been tried to solve these problems. One of the most recent researches which tried to solve these problems is peer-to-peer (P2P) live video streaming over the DHT network. Yet, the availability and the low-latency problems are still the main obstacles of this system because it fully depends on the user device's capacity. In P2P live video streaming if the user device's capacity is poor, other clients will not get the desirable quality of the video stream.

Peer-to-peer systems are distributed systems in which each node has equivalent functionality and without any centralized control or hierarchical organization [1]. The core operation of P2P system is the efficiency of locating each node in the network. The distributed hash table (DHT) is one of the most efficient methods which can be used to locate the node in mesh-based P2P network. DHT is a decentralized system that provides a lookup service similar to a hash table. The DHT node uses key to lookup for the value which associated with it. The advantages of DHT are: high scalability, reliability, and self-organizing [2]. However, only few works have been focused on using DHT with mesh-based mechanism. Thus, cloud assisted live video streaming over DHT overlay network is proposed in this paper.

In this system, the P2P network is based on a DHT using the available protocols such as Chord [1], Pastry [3], Kademlia [4], etc. A cloud server is considered as a node, the same as other users' devices. The only different is that cloud server has a big and stable capacity. The role of the cloud server is to assist other nodes when the numbers of available segments are less than the specific threshold.

The rest of this paper is organized as follow: Section 2 talks about the related works. The system architecture of the new proposed system, buffer map management, data push and pull among peers, and cloud assist scenario are discussed in Section 3. The system performance evaluation is discussed in Section 4. Finally, the conclusion is presented Section 5.

## II. RELATED WORKS

There are many existing researches have been studying on P2P video live streaming; however, most of those works focused on tree-based, mesh-based with gossip protocol and others, and hybrid structure such as [5]-[7]. Only few researches focused on using mesh-based with DHT [2] and cloud assist in P2P live video streaming [8]-[10].

[2] is one of the most recent research which built a P2P live video streaming over DHT network and using chord protocol. This work was proposed to solve the problems of scalability, availability, and low-latency in P2P live video streaming systems. The performance of this system is far better than that of mesh-based and tree-based system. However, it fully depends on user devices' capacity which is small and not stable (nodes join and leave the system any time) and also the video server can become a bottleneck when the number of users is suddenly increased. In order to maintain the video quality and the scalability of the system, and also to deal with the inconsistency of the clients' devices, cloud assist has been proposed in our paper.

[8] proposed a cloud-based service called "AngelCast". AngleCast is proposed to solve the limitation of the delivered video quality. The client nodes in AngleCast are arranged in tree-based structure. [9] proposed Cloud-

 ISBN: 978-1-61208-428-2 

Assisted P2P Live Streaming (Clive) to tackle the bottlenecks in the available upload bandwidth at media source and inside the overlay network that can limit the quality of the server (QoS). Cloud-Assisted Live Media Streaming (CALMS) [10] books and adjusts adaptively cloud server resources in a fine granularity to accommodate temporal and spatial dynamics of demands from live streaming users.

As most of the proposed cloud assisted P2P live video streaming researches tend to solve the problems of tree-based, mesh-based or hybrid structure, only few works are focused on hybrid structure with DHT network. Thus, in this paper, a cloud assisted live video streaming system over DHT overlay network is proposed.

## III. SYSTEM ARCHITECTURE

The cloud assisted video live streaming system is built over DHT overlay network, as shown in Figure 1. In this system, cloud servers join the DHT network as a node the same as other clients and receive video segments directly from the video server.

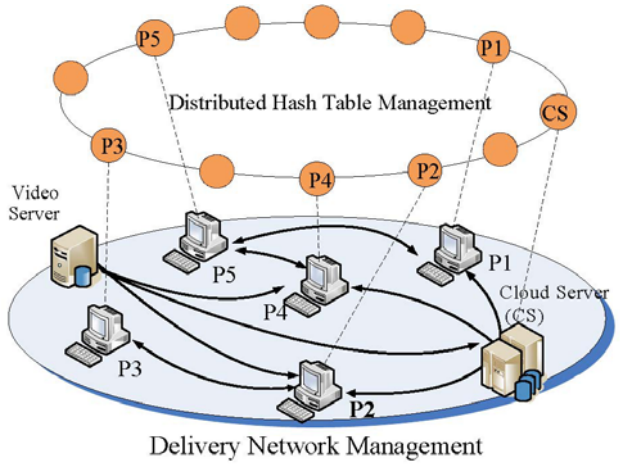


Figure 1. Cloud assisted live video streaming over DHT overlay network

### A. *Buffer Map Management*

In P2P live video steaming system, the video stream is divided into segment or chunk which identified by the sequence numbers, name as segment ID [11]. Each node has a buffer to catch the video segment. A buffer map, which stores several video segments, as shown in Figure 2, is an abstract description of this buffer. Buffer map consists of a sequence of {0, 1} to indicate the buffered and empty states of the segment [12]. While playing the video stream, node can refer to its buffer map in order to know which segment is missing and which segment is already buffered. Thus, node can request the missing segment in advance. In this system, each node stores the buffer map information list of the segment which hash key's ID equals to or immediately succeeds the node's ID.

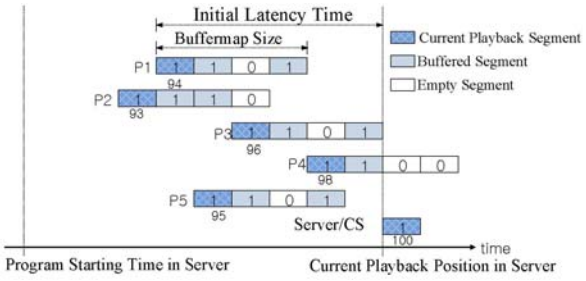


Figure 2. Live streaming scenario

One video segment can have more than one hash key because segment belongs to node in different location will generate different hash key. Hash key contains channel information, program information, segment number and location information, as shown in Figure 3.

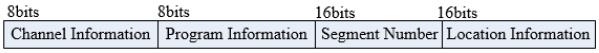


Figure 3. Hash key

The buffer map information consists of the IP address, starting segment and the current buffer map, as shown in Figure 4. The buffer map information list stores the buffer map information in the descending order of the starting segment number.

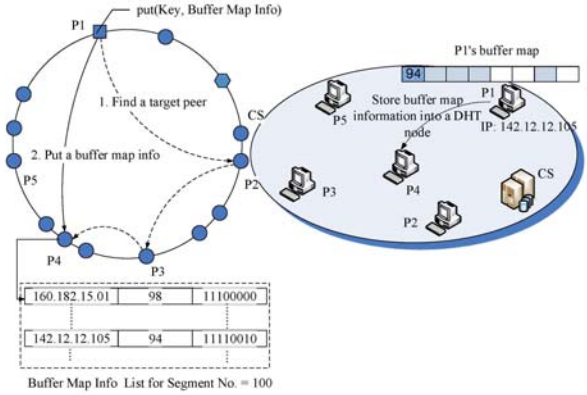


Figure 4. Storing buffer map info process

When node receives segment from server or from other node, this node has to register its buffer map information to the DHT network. The processes of registering its buffer map information are: first, hash key for this segment is created. This hash key is used to find the node to store the buffer map information. The same segment, which has the same hash key, will be stored in the same node. Then, the DHT method, *put* (key, buffer map info), is used to find the appropriate node to store the buffer map information, as shown in Figure 4.

### B. *Data Delivery Management*

During playing the video stream, each node checks their buffer map information in order to know whether there is a

 ISBN: 978-1-61208-428-2 

missing segment. If there is a missing segment, node uses the missing segment's ID, channel information, program information, and node's location information to create a hash key. Then, this hash key is used to find the node which stores the buffer map information list of this missing segment. After that, the requested node requests the buffer map information list from that node. The DHT method, *get* (key), is used to retrieve the buffer map information from specific node, as shown in Figure 5.

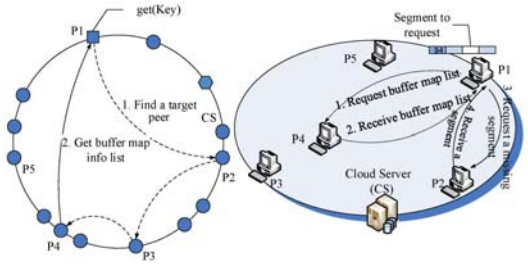


Figure 5. Request segment among peers

Finally, the requested node uses the received buffer map information list to find the node which has the missing segment and sends the request to that node.

### C. *Cloud Server Assist Scenario*

The role of cloud server is to assist other nodes in the DHT network. Cloud server will assist other nodes in case the number of nodes which has the video segment is less than the threshold. The threshold is referred to the number of nodes which receive segment directly from server that can effectively share the video segment to other nodes in the network.

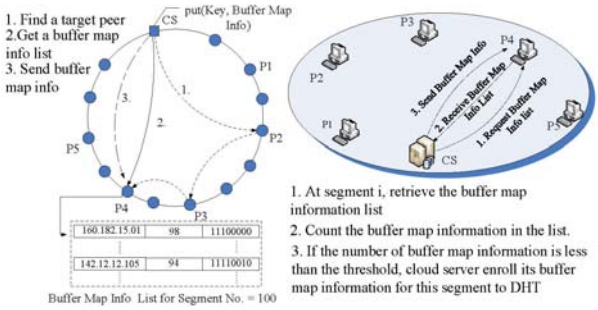


Figure 6. Cloud assist scenario

As illustrates in Figure 6, for each segment, cloud server requests the buffer map information list from the node which stores this list. Then, cloud server uses the buffer map information list to count the available segments. If the numbers of available segments is less than the threshold, cloud server generates hash key and sends its buffer map information to the node which stores the buffer map information list of this video segment. Thus, other nodes can request the video segment from cloud server. After receiving the segment, those nodes also register their buffer map to the network to share the segment to other nodes. Therefore, the cloud server will not be overloaded by many requests.

## IV. PERFORMANCE EVALUATION

In order to evaluate the system performances, a simulation program has been developed. In our experiment, we set the upload bandwidth of the server and cloud to 40,000kb/s and upload and download bandwidth of nodes to 600kb/s. The video size is set to 120 and segment size is 300kb, that one segment can be played for 1 second. Thus, the video length is 2 minutes. The threshold is set to 16 percent of the total nodes. The buffer size is set to 10. After the upload bandwidth of the clouds and the nodes which have the video segment are full, the requested nodes are queued and will request the video segment again after the bandwidth is available. The number of nodes join and leave the network are set to the exponential distribution with mean value of 2 percent and 1 percent of total nodes every half second, respectively. The delay time of each segment is set to 3 seconds, means that if all nodes receive the segment during 3 seconds, the user can watch the video stream smoothly. We run each experiment five times and report the average as the final experimental results.

### A. *Latency*

As illustrates in Figure 7, the average delay time of the segments without the help of cloud assist exceed 3 seconds if compare to when using cloud assist, which only exceed 3 seconds, when the number of nodes reach 3000.

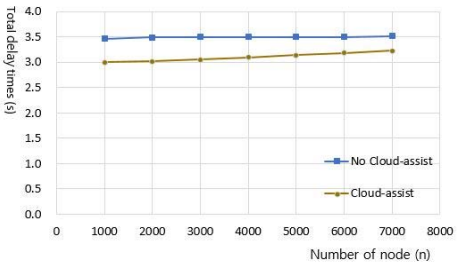


Figure 7. Delay time vs. number of nodes

As mention in the experiment setup, the threshold is set to 16 percent of total nodes. Thus, even when the number of nodes reach 8000, the video server still have enough bandwidth to send the video segment to threshold nodes (the total bandwidth which is required for sending the video segment to the threshold nodes is only 37,500KB). However, because nodes dynamically leave and join the network, the delay time of each segment cannot be maintained. For cloud-assist method, the average delay time of each segment exceed 3 seconds when number of nodes reach 3,000 because cloud server does not have enough bandwidth.

 ISBN: 978-1-61208-428-2 

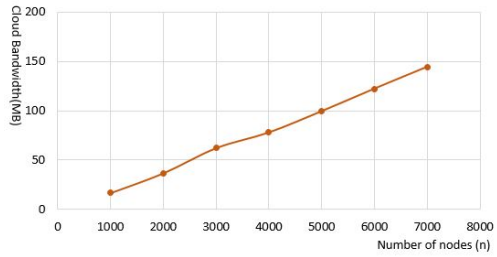


Figure 8. Cloud bandwidth vs. number of nodes

Figure 8 shows the amount of cloud bandwidth which is needed for maintaining the delay time of the network, i.e., if this amount of cloud bandwidth is used for each method, the delay time will not exceed 3 seconds.

### *B. Availability*

Figure 9 shows the number of remained nodes when the number of nodes in the network is increased. The remained nodes refer to the nodes which do not receive the video segment after the delay time (the delay time is set to 3 seconds in this experiment) is passed. The number of remained nodes of no-cloud method is worse than cloud-assist method.

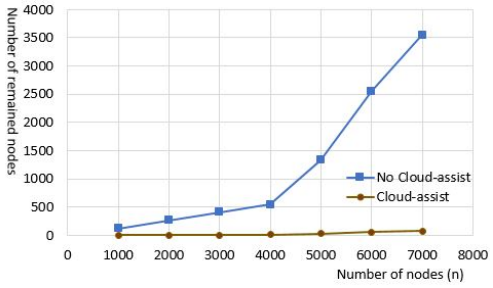


Figure 9. Number of remained nodes vs. number of nodes

Even the delay time, as shown in figure 7, of both methods has a little different, the number of remained methods of no-cloud assist is far bigger than that of cloud-assist methods. For instance, when the number of nodes reaches 3000, the delay time of no-cloud method is 3.49 and the delay time of cloud-assist method is around 3.05. However, according to figure 9, for no-cloud assist, after the delay time 3 (after 3 second) passed, the number of remained nodes is 413 compare to the remained nodes of cloud-assist methods is only around 5.

## V. CONCLUSION AND FUTURE WORK

In this paper, cloud assisted live video streaming over DHT overlay network has been proposed. The DHT network can be used to solve the problem of scalability. Moreover, because cloud server is consistent, has big capacity, and receives video segments directly from video server, the availability of the video segments and the low-latency can be enhanced. The performance evaluation results show that with the present of cloud assist, the performances of the system are greatly improved in both availability and low-latency. We only build a simulation program to compare the performance of the system when there is no cloud assist and when there is a cloud assist. In the future, a full function DHT simulation process should be conducted. In addition, the threshold value and the amount of cloud capacity which is required for assisting the failure nodes also need to be studied in details in order to maximize the system performance and minimize the cost.

## REFERENCES

[1] I. Stoica, et al., “Chord: A scalable peer-to-peer lookup protocol for internet applications,” IEEE/ACM Trans. Networking, vol. 11, no. 1, 2003, pp. 17-32.

[2] H. Shen, Z. Li, and J. Li, “A DHT-aided chunk-driven overlay for scalable and efficient peer-to-peer live streaming, IEEE Trans. on Parallel and Distributed Systems, vol. 24, no.11, 2013, pp. 2125-2137.

[3] A.I.T., Rowstron, P. Druschel, “Pastry: Scalable, decentralized object location and routing for large-scale peer-to-peer systems,” Proc. of the 18th IFIP/ACM International Conference on Distributed Systems Platforms, 2001, pp. 329-350.

[4] P. Maymounkov, D. Mazieres, “Kademlia: a peer-to-peer information system based on the XOR metric,” 1st International Workshop on Peer-to-peer Systems, 2002.

[5] F. Picconi and L. Massoulie, “Is there a future for mesh-based live video dtreaming?” Proc. 8th Int’l Conf. Peer-to-Peer Computing (P2P), 2008, pp. 289-298.

[6] J. Venkataraman and P. Francis, “Chunky spread: Multi-tree unstructured peer-to-peer multicast,” Proc. 5th Int’l Workshop Peer-to-Peer Systems (IPTPS), 2006, pp. 1-10.

[7] F. Wang, Y. Xiong, and J. Liu, “mTreebone: A hybrid tree/mesh overlay for application-layer live video multicast,” Proc. 27th Int’l Conf. Distributed Computing Systems (ICDCS), 2007.

[8] R. Sweha, V. Ishakian, and A. Bestavros, “AngelCast: Cloud-based peer-assisted live streaming using optimized multi-tree construction, Proc. 3rd Multimedia Systems Conference (MMSYS), 2012, pp. 191-202.

[9] A. H. Payberah, H. Kavalionak, V. Kumaresan, A. Montresor, and S. Haridi, “CLive: Cloud-assisted P2P live streaming,” Proc. 12th IEEE Int’l Conf. Peer-to-Peer Computing (P2P), 2012, pp. 79-90.

[10] F. Wang, J. Liu, and M. Chen, “CALMS: Cloud-assisted live media streaming for globalized demands with time/region diversities,” IEEE Infocom, 2012, pp. 199-207.

[11] Y. Zhou, D. M. Chiu, and J. C. Lui, “A simple model for analyzing P2P streaming protocols,” IEEE Int’l Conf. Network Protocols (ICNP), 2007, pp. 226-235.

[12] S. Lan, S. Zhang, X. Zhang, and Z. Guo, “Dynamic asynchronous buffer management to improve data continuity in P2P live streaming, IEEE 3rd Int’l Conf. Computer Research and Development (ICCRD), 2011, pp. 65-69.

 ISBN: 978-1-61208-428-2 