A CDN-P2P Hybrid Architecture with Location/Content Awareness for Live Streaming Services

Kim-Thinh Nguyen and Young-Han Kim
School of Electronic Engineering, Soongsil University
Dongjak-Gu, Seoul, 156-743, Korea
[e-mail: {nkthinh, yhkim}@dcn.ssu.ac.kr]
*Corresponding author: Young-Han Kim

Received August 31, 2011; revised October 18, 2011; accepted October 26, 2011; published November 29, 2011

Abstract

The hybrid architecture of content delivery networks (CDN) and peer-to-peer overlay networks (P2P) is a promising technology enables effective real-time streaming services. It complements the advantages of quality control and reliability in a CDN, and the scalability of a P2P system. With real-time streaming services, however, high connection setup and media delivery latency are becoming the critical issues in deploying the CDN-P2P system. These issues result from biased peer selection without location awareness or content awareness, and can lead to significant service disruption. To reduce service disruption latency, we propose a group-based CDN-P2P hybrid architecture (iCDN-P2P) with a location/content-aware selection of peers. Specifically, a SuperPeer network makes a location-aware peer selection by employing a content addressable network (CAN) to distribute channel information. It also manages peers with content awareness, forming a group of peers with the same channel as the sub-overlay. Through a performance evaluation, we show that the proposed architecture outperforms the original CDN-P2P hybrid architecture in terms of connection setup delay and media delivery time.

Keywords: Peer-to-peer network, content delivery network, live streaming services, content awareness, location awareness

A preliminary version of this paper appeared in ISCE 2011, June 14-17, Singapore. This version includes a mechanism to handle churn-rate problem and extend simulation by OMNeT++ to obtain results on live streaming service networks. This research is supported by the Ubiquitous Computing and Network (UCN) Project, Knowledge and Economy Frontier R&D Program of the Ministry of Knowledge Economy(MKE) in Korea as a result of UCN's subproject 11C3-C1-20S, and the IT R&D program of MKE/KEIT [KI001822, Research on Ubiquitous Mobility Management Methods for Higher Service Availability]

DOI: 10.3837/tiis.2011.11.015

1. Introduction

Hybrid content delivery networks (CDN) and peer-to-peer (P2P) overlay networks are complementary technologies to facilitate large-scale and reliable media distribution with low deployment cost. Thus, the CDN-P2P hybrid architecture has been highly recommended as an effective real-time streaming approach [1]. In related works, several CDN-P2P hybrid architectures have been studied. Dongyan Xu et al. [2] have proposed a cost-effective CDN-P2P mechanism to obtain handoff time between CDN-P2P and P2P, reducing the management cost of CDN servers. Cahill et al. [3] suggested a novel algorithm relating to the effective placement of CDN servers for high-quality media delivery. These studies primarily cover certain integration and collaboration issues between CDN and P2P.

To provide a seamless real-time streaming service, the peer selection mechanism is the crucial concern among several P2P issues. In the original CDN-P2P, a random peer-selection mechanism is used to lookup available target peers regardless of their physical location. This increases the connection setup time (first playback delay) and packet delivery latency considerably between the requesting peer and the target peer, and eventually, affects the seamless media streaming services.

Until now, the issue of biased peer selection in the CDN-P2P architecture has not been addressed. This gap motivated us to propose a group-based CDN-P2P architecture (iCDN-P2P), which creates a location/content-aware peer selection. Specifically, a SuperPeer network makes a location-aware peer selection by employing a content addressable network (CAN), which has the advantage of location awareness to distribute channel information. It also manages peers with content awareness, forming a group of peers with same channel as the sub-overlay.

In the iCDN-P2P, a new peer sends a channel request to a SuperPeer located in the same CAN zone to obtain a peer list of its channels of interest. The SuperPeer provides the peer with a list of those peers that are physically close to the requesting peer, which then connects with the peers obtained in the list. This can significantly decrease connection setup time compared to the original CDN-P2P architecture. Moreover, it greatly reduces media delivery latency to forward streaming packets.

To ensure dependability of the overlay in case of a high churn rate, peers with neighbor links to the joining or leaving peers require updates. As the churn rate increases, the overlay network eventually partitions, causing lookup queries to return inconsistent results and significant degradation in the overlay's service quality. Voulgaris et al. [4] described a gossip-based membership management protocol, CYCLON, which deal with a high node churn in unstructured P2P overlays. CYCLON acts as a lightweight protocol, meaning that the peer obtains random nodes of the network to execute gossiping. This random peer selection in the gossip protocol is an inefficient strategy to maintain the overlay. Because a particular peer can be chosen to execute gossip many times, that incurred unnecessary maintenance delay. In this paper, we modify gossip protocol, which based on a heuristic approach, to handle churn-rate in our specific architecture.

The remainder of this paper is organized as follows. First, the related studies are described, second, a solution and related data structure are presented. Third, churn rate handling is proposed in the next section. The performance analysis in both static and dynamic environment is described in section 4. Finally, concluding remarks are offered.

2. Related Work

2.1 Current P2P Live Streaming Systems

Nowadays, several P2P streaming systems have adopted a mesh-based streaming approach and have deployed successfully in the real environment, such as PPTV [5], PPStream [6], and CoolStreaming [7]. These technologies offer high scalability and reliability compared to the tree-based approach, when the number of participating peers increases.

Fig. 1 shows a general P2P live video streaming network. A new peer joins the overlay and gets a live streaming channel; first, it contacts the tracker server to obtain a list of peers that are watch the same channel (step 1). The tracker returns a number of active peers (step 2). The peers establish connections with one other and exchange a peer list, chunk bitmaps information in a neighbor list, and obtain video content (steps 3 and 4). Finally, each peer must report its status, chunk information to the tracker server periodically (step 5).

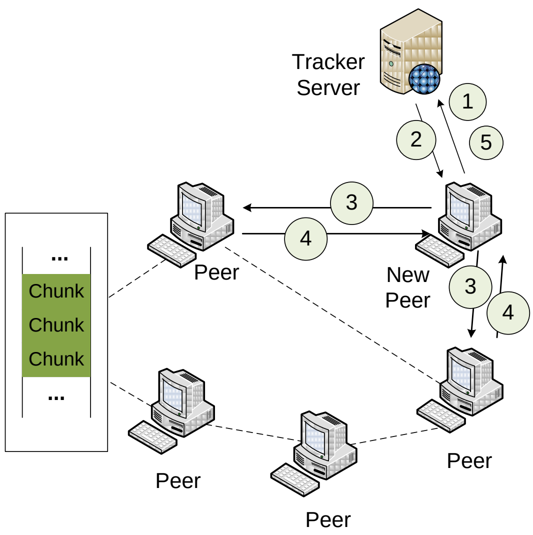


Fig. 1. General P2P live streaming architecture

2.2 The existing CDN-P2P Hybrid Network Model

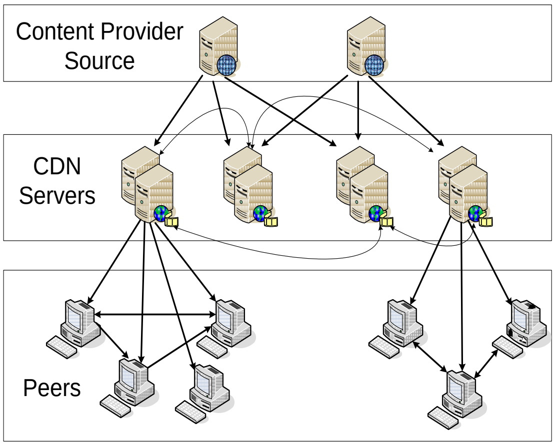


Fig. 2. Network model of CDN-P2P hybrid architecture

The existing architecture of CDN-P2P [8] consists of a two-level hierarchical model, as shown in **Fig. 2**. The process of content delivery is divided into two phases: CDN distribution at the top, and P2P distribution at the bottom. The content source disseminates data to dedicated CDN servers, and caches there. A new peer sends a request to the nearest CDN server to obtain data as quickly as possible. After that, another peer wants to obtain the same content; it contacts the peers that are inside its P2P overlay to exchange content with each other.

3. A Group-based CDN-P2P Hybrid Network Architecture

3.1 Group-based CDN-P2P Hybrid Network Architecture (iCDN-P2P)

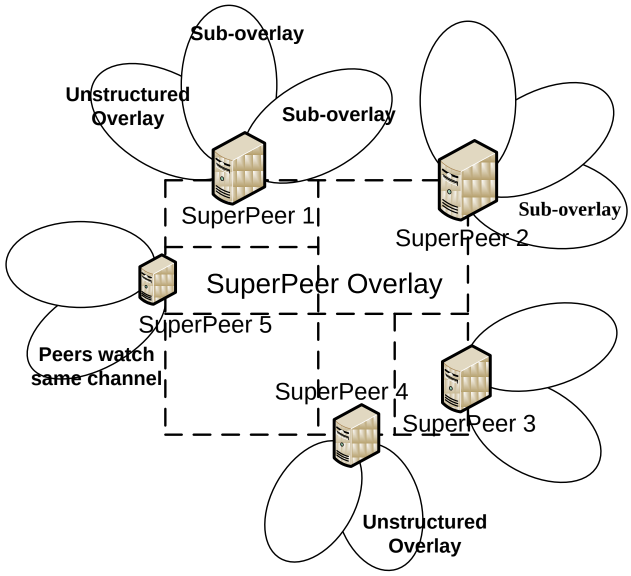


Fig. 3. The concept of group-based CDN-P2P (iCDN-P2P) hybrid architecture

In iCDN-P2P, a peer that is physically close to a CDN server is selected to act as a SuperPeer, which stores the peer-id lists, the channel information that a peer watches, and peer status information. SuperPeers join a distributed hash table (DHT) overlay to distribute this information. CAN algorithm [9] is applied for building a SuperPeers overlay network. CAN algorithm used a virtual coordinate space (usually 2-dimensional) to store pairs of (key, value), key K is hashed and mapped onto a point T in CAN space. The (K, value) pair is stored at the node that owns the zone which corresponds to the point T. To get a value corresponding to key K, peer looks up the point T which maps key K and then retrieves data from the peer whose zone has point T. Every node in CAN overlay maintains a routing table that holds IP address and virtual coordinate zone of its neighbors. The CAN overlay construction mechanism allocates peers to zones at random and it did not consider peer's location. So, using the binning technique [10], SuperPeers are assigned to different bins, and each bin is allocated to certain zones of the CAN P2P overlay network [9].

Fig. 3 describes the proposed architecture which includes two hybrid layers of a structured and unstructured P2P overlay network:

- SuperPeer overlay: each bin (same area) will select a peer to act as SuperPeer which has functions similar to the tracker in a general P2P live streaming system; these SuperPeers distribute within the overlay instead of a centralized server. The SuperPeer overlay is constructed using a CAN algorithm, and considering location-awareness.

- Sub-overlay (unstructured overlay): at each CDN server's location, peers watching the same channel are grouped as sub-overlay. So, peers that are in the same group have two meanings: they are adjacent in geographical terms, and share an interest in the same video content.

Peers that watch the same channel and that are located in the same physical area form a group sub-overlay. SuperPeer manages every sub-overlay to provide the live video streaming service network.

When a peer wants to obtain peer-lists to make connection with the requesting channel, it contacts the nearest SuperPeer instead of querying to a tracker server. If SuperPeer has no related peer lists for the requesting channel, the peer sends the request to a SuperPeer located in the other sub-overlay within the CAN overlay network.

3.2 Overlay Construction for SuperPeers

In this section, to construct an overlay network for SuperPeers, a binning technique with locality-awareness has been designed. The main idea behind the binning technique is as follows: As a first step, a peer measures its round-trip time (RTT), which is distinguished as a set of pre-defined landmarks; then, we order RTT values by increasing latency. Peers who have the same order form a group within the same bin.

The content addressable network (CAN) overlay construction mechanism allocates peers to zones at random. Thus, adjacent peers on CAN are not topologically close in the underlying IP network. This leads to inefficient routing, because every application-level service on the CAN overlay could be between two geographically-distant peers. Applying this binning strategy, we can construct a CAN overlay network that is congruent with the underlying IP topology.

In the iCDN-P2P architecture, we consider the location of the SuperPeer based on CDN servers when constructing a DHT overlay network. Hence, an improved CAN overlay network is the optimization method to construct a SuperPeer overlay network. SuperPeers will be located in CAN zones according to the order of RTT values achieved.

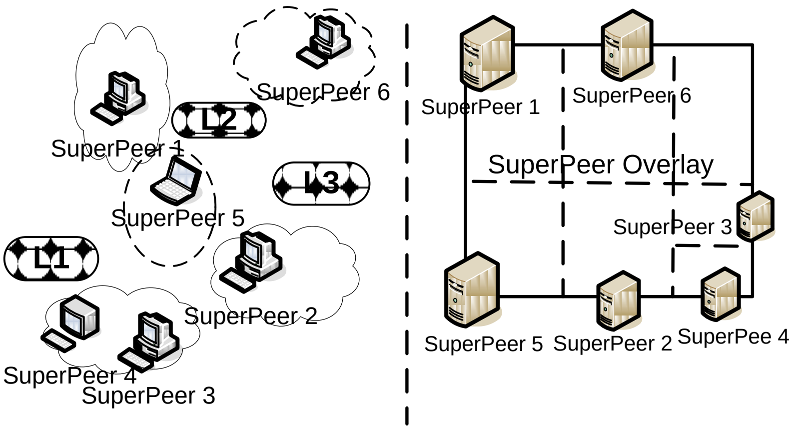


Fig. 4. CAN-based overlay construction for SuperPeers

Fig. 4 provides an instance of SuperPeer overlay construction. If we assume that there are three pre-defined landmarks, we get 3! = 6 bins with maximum counts. These landmarks are applied to calculate RTT to find suitable CDN servers; therefore, the location of the CDN server is also considered. Then SuperPeers are located in CAN zones according to the ordering of RTT values achieved.

When the RTT measurement is inaccurate, the location of SuperPeer in CAN overlay network maybe is not optimal. The allocation of SuperPeers into CAN bases on the order of RTT values achieved. If we increase the number of landmarks, we can reduce the change of this ordering. Therefore, the efficient of SuperPeers' location-awareness property can still achieve.

3.3 System Operation

Fig. 5 shows the operation of a new peer X that wants to get peer lists for channel A. Peer X measures the RTT to each landmark, L1, L2, L3. Then, the values are arranged in ascending order to determine to which bin the peer X belongs. From the proposed signaling procedure, peer X determines a suitable SuperPeer that is located in the same area (e.g., SuperPeer 1). It sends a request to SuperPeer 1 to obtain the peer lists related to channel A.

The following steps describe the system operation in detail:

***Step 1*:** If there are peers that watch channel A, SuperPeer 1 returns a peer list (P1, P2, P3) according to a peer-list request.

***Step 2*:** If there is nothing related to channel A, peer X sends a request() signal to SuperPeer 1 to obtain the hashed values for channel A. In this case, SuperPeers 3 and 4 are returned. If the number of peers watching channel A is larger than the other one, peer X sends a request() signal to SuperPeer 4 by choosing SuperPeer1.

Periodically, the other peers update their status information to SuperPeer; the update message includes information about peer status, regarding peers left already or still alive.

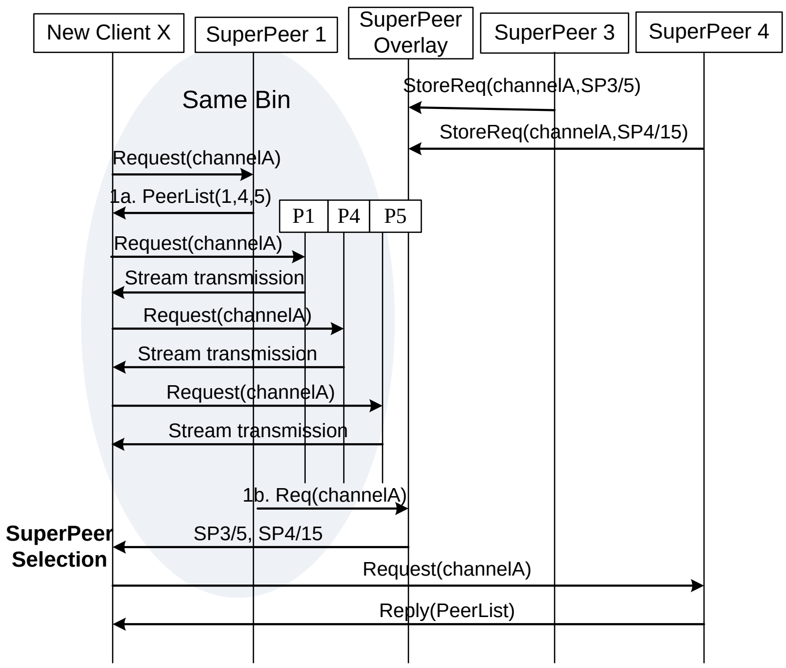
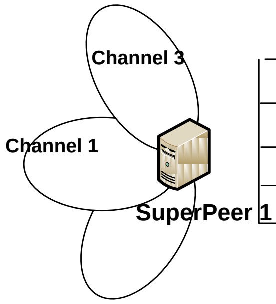


Fig. 5. Content/location-aware peer selection mechanism

3.4 Data Structure



Channel-ID	Peer-ID	Status
Channel-1	Peer-1	Alive
Channel-2	Peer-1	Alive
Channel-3	Peer-1	Left

Fig. 6. Sub-overlay structure in details

A SuperPeer maintains a database for the peers located at the same sub-overlay that corresponds to each channel. Each peer inside a sub-overlay adds an entry into database consisting of 3 fields as in **Fig. 6**:

- Information about the channel existing at the considered location
- List of peer IDs, which denote hashed values of IP address and port number
- Statuses of each peer for failure and leave detection

When a peer sends a request to get a video channel in the DHT overlay network, a list of SuperPeers is returned. An efficient SuperPeer selection is determined by defining a new data storage mechanism for a CAN algorithm. The live streaming service network in this storage situation requires two main functions: registration and lookup. SuperPeers store the mapping from the channel-ID to the SuperPeer-ID attaching the number of peers that are currently watching the channel.

mapping(Channel-ID, SupeerPeer-ID + Number of Peers)

Lookup: a new peer (client function) uses the *fetch()* method of data storage to retrieve the SuperPeer-ID that had registered the channel information.

To register channel information, a peer stores the PStreamRegistrationData structure. For example, SuperPeer 1 has obtained channel information about “channel A” whose channel-ID is “Channel 1”, and includes the number of peers in sub-overlay that are watching channel A, called the peer list.

An instance of the stored mapping in the CAN overlay:

1. Channel-ID1 --> SP1, 3, [peerID4, peerID3, PeerID9]
2. Channel-ID1 --> SP3, 5, [peerID1, peerID2, peerID5, peerID8, peerID7]

Fig. 7. An instance of stored data in SuperPeer 1

The contents of a PStreamRegistrationData structure are described as in **Fig. 8**:

- PeerID: only local meaning inside sub-overlay
- PStreamRegistration Kind-ID definition includes: Kind-ID is the Channel-ID. The stored data are a PStreamRegistrationData, which contain the combination of the SuperPeer-ID and PeerID-list (peers interested in the same channel).

```
1. struct{
2. uint16 peer_id_list_length;
3. PeerID peer_id_list [peer_id_list_length];
4. }DestPStreamData
5. struct{
6. opaque superPeer_ID [2];
7. DestPStreamData data;
8. }PStreamRegistrationData
```

Fig. 8. PStreamRegistrationData Pseudo Code

According to the lookup function, a peer uses the *request()* method to retrieve SuperPeer-ID and peerID-list. Based on these peerID-lists, a peer can decide which SuperPeer is the most suitable. Finally, the peer establishes connections with peers within the peer list.

3.5 Churn Rate Handling

To ensure dependability of the overlay in case of a high churn rate, peers with neighbor links to the joining or leaving peers require updates. As the churn rate increases, the overlay network eventually partitions, causing lookup queries to return inconsistent results and significant degradation in the overlay's service quality.

Dealing with the highly dynamic nature of peers is crucial to ensure the robustness of the P2P-CDN. In our proposed architecture, there are two situations that cause a churn-rate problem:

- Peers in the sub-overlay join or leave
- SuperPeer leaves

In this section, we focus on the procedures that maintain sub-overlay by detecting failure peers as quickly as possible.

3.5.1 Churn Rate Handling caused by Regular Peers in Sub-overlay

When regular peers (peers in sub-overlay) leave or fail frequently, the existing peers have to connect with other peers to establish new connections and get video chunks. There is a playback latency and video transmission delay of a few minutes in case of a dynamic environment. To reduce this latency and delay, a gossip protocol [11] can be used to maintain the P2P overlay. However, the peer selection for gossiping in this protocol is random. That caused inefficiencies when updating peers' status information among one another.

To optimize the gossip protocol for solving the churn-rate problem in our architecture, we propose a new mechanism based on the heuristic method to modify the gossip protocol. This method adds new field to exchanged messages to set peer priority. Instead of selecting peers randomly for gossiping, a peer with a higher time to live (TTL) will be selected. The reason is that a peer that joins to the overlay first and has no updates from other peers will have a high probability failure. To do that, each peer maintains a new value – *Time to Update (TTU)* – denoting the time since the moment it was updated by the other peer.

When a peer executes a gossip procedure actively, active behavior can be described as in **Algorithm 1**. Periodically, after *Tgossip* (gossip period), peer[a] initiates a gossip exchange. Peers in the peer[a]'s view will increase their own TTU value, meaning that the period that the peers have not updated via gossip protocol by other peer yet. Churn information of all peers in peer[a]'s view is stored in *churn_info*. Peer[a] selects the peer with the highest TTU for updating by gossip exchange protocol. Then, the TTU value of peer[b] is reset to zero.

```
wait(Tgossip)
for(int i=1; i<viewSize; i++)
view[i].TTU++;
peer[b] <-- peer[a].select_oldest();
peer[a].send(GossipMess) --> peer[b]
peer[a].receive(GossipMess') <-- peer[b]
set(peer[b].TTU) --> 0
```

Algorithm 1. Active behavior of peer[a]'s improved gossip protocol

3.5.2 Churn Rate Handling caused by SuperPeer

When the SuperPeer leaves, it is important to back up the management information to another SuperPeer to maintain the operation of the whole system. The SuperPeer's CAN structure overlay provided a mechanism to address the churn-rate problem. When a peer in the CAN overlay leaves or fails, another peer, one that has the nearest zone will cover its function in overlay. So, before the SuperPeer in our proposed architecture leaves the overlay, these peers back all its information up to the other peer. Simultaneously, SuperPeer notifies all the peers in its sub-overlay about its leaving by broadcasting a message.

Fig. 9 is an instance of maintenance mechanism when SuperPeer 2 leaves the overlay. We assume SuperPeer 4 is allocated on the nearest zone with SuperPeer 2. SuperPeer 2 backs all its information up to SuperPeer 4 before leaving the overlay and SuperPeer 2 notifies to all peers in its sub-overlay.

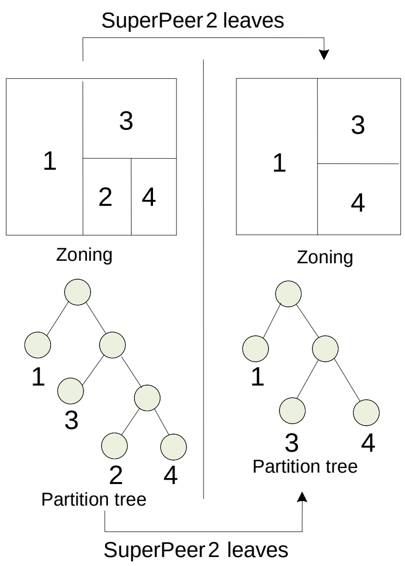


Fig. 9. Maintenance mechanism when SuperPeer leaves

In case of a SuperPeer failure, a virtual SuperPeer is designed to support this issue.

SuperPeer backs its information up to a virtual SuperPeer, and this virtual SuperPeer is administered by the operator.

4. Simulation Evaluation

4.1 Simulation Scenario

We conduct simulation-based experiments using the OverSim framework [12] over OMNeT++ v4.1 [13]. In order to simulate a CDN-P2P hybrid network for providing live streaming services, an overlay with the following functionalities is created:

- Mesh construction & maintenance
- Packet scheduling
- Buffer map exchange & video transmission

In application layer, Tier 1 provides a mechanism to schedule video frame, to exchange neighbor's buffermap. At Tier 2, the CDN server will stream video to regular peers by attaching a Camera function. Encoded MPEG-4 video files are distributed here. The camera function reads this video trace file to send to requested peers. (An MPEG-4 file can be downloaded from Arizona University: *http://trace.eas.asu.edu*). A Player function is attached to regular peers to decode MPEG-4 video.

Fig. 10 describes a scenario of our proposed CDN-P2P hybrid network architecture. First, we initiate one Tracker server for providing and updating a peer-list to new peer, and five CDN servers to distribute live video streaming to regular peers.

The number of peers is initiated at 100 peers, the interval between node creation is 0.1 seconds, meaning that the creation of new peer is ready to join the overlay to receive live streaming 0.1 second after the previous peer was generated.

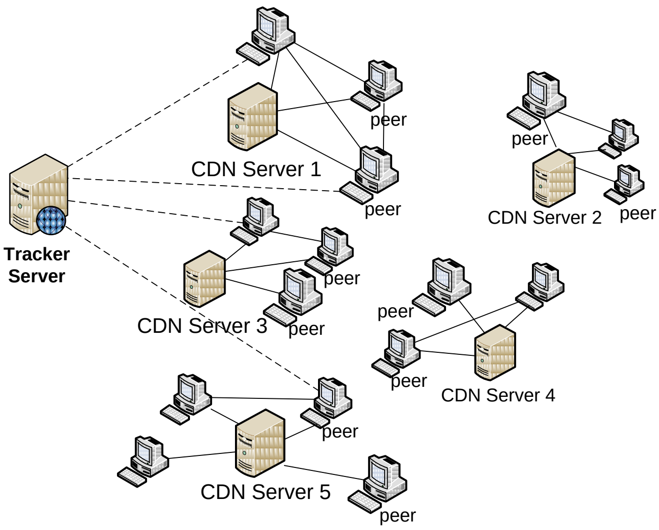


Fig. 10. Proposed CDN-P2P hybrid network scenario (iCDN-P2P)

When a new peer joins the overlay network, it contacts the tracker server to get a peer list. At the beginning, the tracker chooses the CDN server randomly and returns to the new peer. Peers establish connections with the CDN server and other peers in a received peer list. For simulating our proposed CDN-P2P hybrid architecture, peers establish five groups that correspond to the five CDN servers' location. In a real situation, the CDN servers are distributed accross the world; therefore, the distance between CDN servers is much larger. Also, new peers will determine the nearest CDN server based on some algorithms in the CDN network, which is not random, as in this simulation.

Fig. 11 shows the operational procedure of our proposed CDN-P2P hybrid architecture, iCDN-P2P.

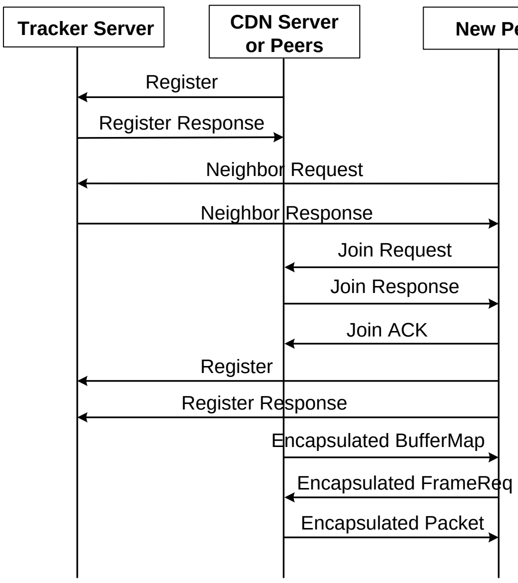


Fig. 11. Operational procedures

Fig. 12 describes an instance of an original CDN-P2P hybrid network architecture (oCDN-P2P) which Xu et al. have described [2]. In this architecture, when a new peer requests a list of peers, the tracker returns peers randomly.

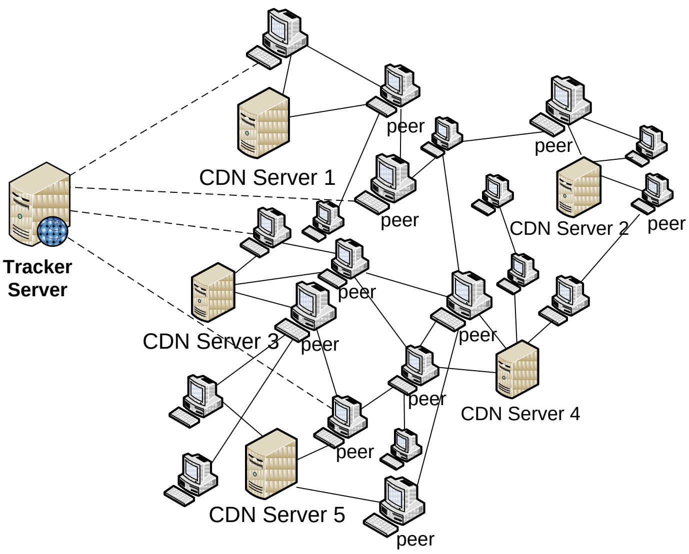


Fig. 12. Original CDN-P2P hybrid architecture as in [2]

Other parameters are initialized in this simulation:

Table 1. Parameters configuration

Parameter Name	***Value***
Packet size	1000 byte
Source bandwidth	6 Mbps
Video average rate	512 Kbps
Frame per second (*)	25
Storage capacity in peer buffer (*)	30s

(*) Camera/Player's Parameters

4.2 Simulation Results

4.2.1 Simulation Results in Static Environment

In the simulation, we evaluate the improvement of our proposed CDN-P2P hybrid architecture (iCDN-P2P) compared to the original CDN-P2P architecture (oCDN-P2P), in terms of the following factors:

- End-to-end delay: the time between creating a frame in the source node and playing it in destination node, and the average value for which each peer receives all frames in simulation time
- Frame error rate: determined by total frames loss/total frames x 100
- First playback delay: the time from when a peer joins the overlay and requests the video frame until the peer receives the first frame to play

The simulations are executed within 200 seconds, five times for each simulation. Then, we obtain statistical information by "*globalStatistics* → *addStdDev*", then we calculate the average value and represent output to Gnuplot [14].

The results of two architectures, in terms of end-to-end delay, are described in **Fig. 13**. This metric becomes lower for improved CDN-P2P architecture (iCDN-P2P) as network nodes increase. Peer selection with location awareness reduces the distance for video transmission between peers, and with content awareness, can extend the list of peers who are able to watch the same channel easily. So, when the network grows, the difference of two architectures is clear.

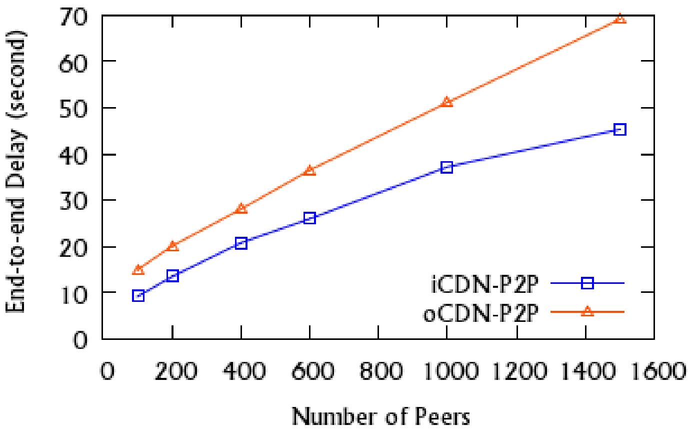


Fig. 13. Average end-to-end delay against the number of peers joins the overlay

Fig. 14 shows the frame error rate, or called as frame lost. For simulating live video streaming, the structure of video trace file includes *Time* field. This value indicates the timestamp which receiver has to receive video packet. We can determine the number of frame loss based on this timestamp. If a peer receives the frame which does not meet the requirement of time field, that is an error frame. In Fig. 14, when the number of peers increases, both architectures raise the frame error rate up. Among that, our proposed architecture with location/content-awareness reduced the frame loss. The distance between two peers in our architecture is shorter than the original architecture, so it takes a shorter period to transfer video.

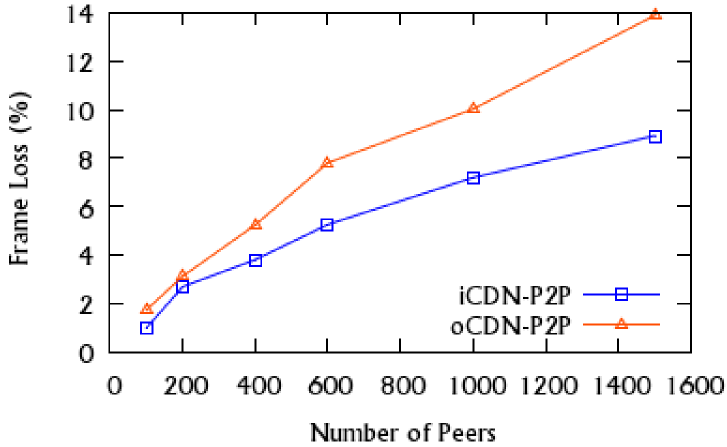


Fig. 14. Frame error rate of received video against the number of peers

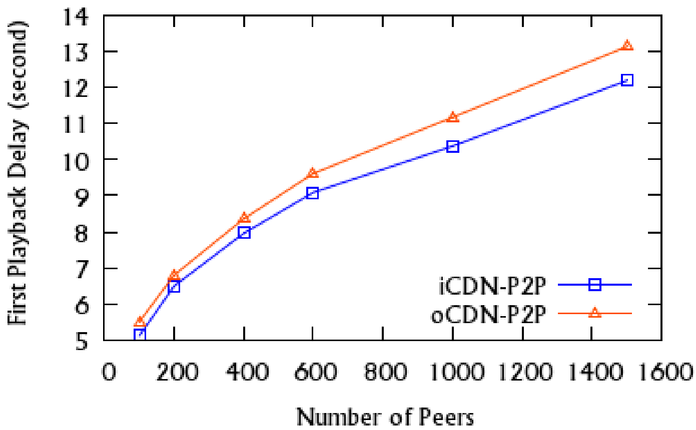


Fig. 15. First playback delay against the number of peers

Fig. 15 shows the first playback delay as the network grows. Because of the limited buffering capacity of CDN servers and peers, at a particular time, the number of frames that are provided to new peers is also limited. When the number of peers increases, those peers who joined late have to wait for a longer period to start playing the video. In our architecture, the CDN server returns a list of peers considering the peers' location (peers that are closest to requesting peer), so the delay for video frame transmission will be reduced.

4.2.2 Simulation Results in Dynamic Environment

In the previous simulation, there are three kinds of peers: the Tracker server *(*.churnGenerator[0])*, CDN server *(*.churnGenerator[1])*, and regular peers *(*.churnGenerator[2])*. All peers are simulated with noChurn, meaning that we simulate our scenarios in a static environment.

For simulating in a dynamic environment, we use the LifetimeChurn in Oversim. The lifetimeMean = 5 (s), which means that after five seconds a random peer leaves the overlay.

As with operation of an original CDN-P2P hybrid architecture, peers in overlay will periodically send a Notification_Neighbor message to the tracker server to update the information about its neighbors. If the interval time for sending this message is too short, the overhead of the network will increase and consume processing resources for unnecessary messages, because the peer list in each peers did not change over a short period.

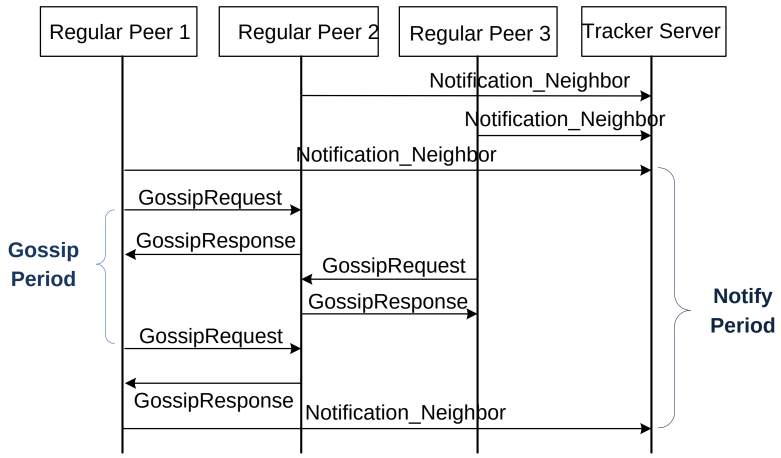


Fig. 16. Improved gossip protocol operation in simulation

The peer receives a Neighbor_Response message from the tracker server, using an array to store the peer list. So, after joining the overlay and requesting the list of peers who are watch the same video streaming, every peer in the overlay system will store the neighbor list. This neighbor list is for implementing the improved gossip protocol, and each neighbor attaches a value; namely, *TTU*.

Periodically (after *Tgossip*), each peer in overlay will create a GossipRequest message to send to one of its neighbors in the neighbor array. This gossip message will update the neighbors' lists among one another. If one peer in the neighbor list fails or leaves, the gossip message updates this information to the passive gossip peer.

In addition, the peer receives GossipRequest message replies GossipResponse message.

In **Fig. 16**, regular peer 1 sends Notification_Neighbor message to tracker server for updating its neighbor information. Periodically (after notify period), it sends this message again. Besides that, regular peer 1 uses GossipRequest message to execute improved gossip protocol with regular peer 2. We assume regular peer 2 is the peer which has the highest TTU.

Fig. 17 is the operation when peer receives Gossip_Request message.

```
if (simpleMeshmsg->getCommand() == GOSSIP_REQUEST)
{
update neighbor list
SimpleMeshMessage* joinResponse = new SimpleMeshMessage("joinResponse");
joinResponse->setCommand(JOIN_RESPONSE);
joinResponse->setSrcNode(thisNode);
sendMessageToUDP(simpleMeshmsg->getSrcNode(),joinResponse);
}
```

Fig. 17. Process of replying GossipResponse message

Fig. 18 provides the results that compare our proposed CDN-P2P hybrid architecture in both a static environment and a dynamic environment. The results show that the CDN-P2P hybrid architecture has been affected by peer churn rate in terms of end-to-end delay.

We simulate a improved gossip protocol in our proposed CDN-P2P hybrid architecture for providing live streaming. The value of updating the neighbor list to the tracker server in both scenarios is the same (value of *neighbor_notify_period*). The results show that the end-to-end video transmission delay, in the case of the executing the improved gossip protocol, is reduced compare to the previous one. The reason for this improvement is the peers in overlay updating each other, so that there is quick notification regarding the leaving or failing of peers.

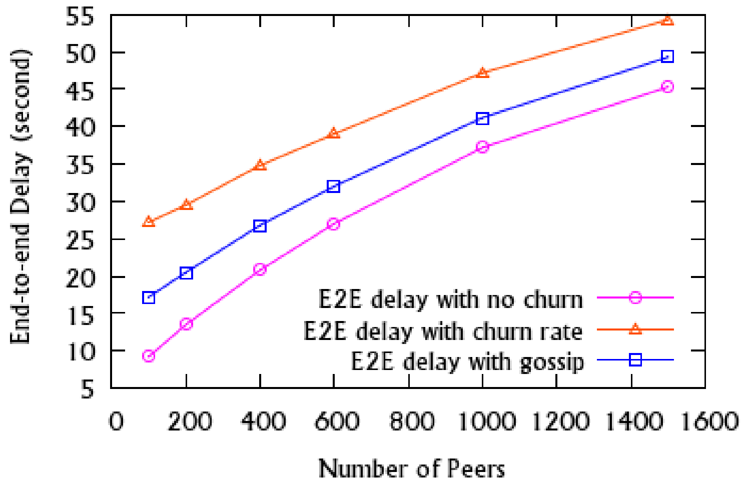


Fig. 18. Simulation results of CDN-P2P hybrid architecture for live video streaming services in dynamic environment

5. Conclusions

A hybrid content delivery network (CDN) and peer-to-peer (P2P) network are complementary technologies that facilitate large-scale and reliable media distribution with low deployment costs. However, in real-time streaming services, the issues of high connection setup and media delivery time have not been resolved. This significant service disruption is caused by biased peer selection without location awareness or content awareness. To reduce service disruption time, as occurred in the original CDN-P2P operation, we proposed a group-based CDN-P2P hybrid architecture (iCDN-P2P), which creates a location/content-aware peer selection. Specifically, a SuperPeer performs a location-aware peer selection by employing a content addressable network (CAN) that has the advantage of location awareness to distribute channel information. It manages the peers with content awareness, forming a group of peers that use the same channel as the sub-overlay uses. The simulation demonstrated that the iCDN-P2P is more efficient than the original CDN-P2P in terms of total service disruption time.

References

[1] C. Huang, A. Wang, J. Li, Keith W. Ross, "Understanding Hybrid CDN-P2P: Why Limelight Needs Its Own Red Swoosh," in *Proc. of 18th Int. Workshop on Network and Operating Systems Support for Digital Audio and Video*, pp. 75-80, May 28-30, 2008. Article (CrossRef Link)

[2] D. Xu, S. Kulkarni, C. Rosenberg, H. Chai, "Analysis of a CDN-P2P Hybrid Architecture for Cost-effective Streaming Media Distribution," *Springer Multimedia Systems*, vol. 11, no. 4, pp. 383-399, Mar. 17, 2006. Article (CrossRef Link)

[3] A. Cahill, C. Sreenan, "An efficient CDN Placement Algorithm for the Delivery of High Quality TV Content," in *Proc. of 12th Annual ACM International Conference on Multimedia*, pp. 975-976, Oct.10-16, 2004. Article (CrossRef Link)

[4] S. Voulgaris, D. Gavidia, M. van Steen, "CYCLON: Inexpensive Membership Management for Unstructured P2P Overlays," *Jounal of Network and Systems Management*, vol. 13, no. 2, pp. 197-217, July 3, 2005. Article (CrossRef Link)

[5] PPTV, http://www.pptv.com/.

[6] PPStream, http://www.ppstream.com/.

[7] X. Zhang, J. Liu, B. Li, T. Peter Yum, "CoolStreaming/DONet: A Data-Driven Overlay Network for Peer-to-Peer Live Media Streaming," in *Proc. of IEEE International Conference on Computer Communications (Infocom)*, pp. 2102-2111, Mar. 13-17, 2005. Article (CrossRef Link)

[8] H. Yin, X. Liu, T. Zhan, V. Sekar, F. Qiu, C. Lin, H. Zhang, B. Li, "Design and Deployment of a Hybrid CDN-P2P System for Live Video Streaming: Experiences with Livesky, " in *Proc. of 17th ACM International Conference on Multimedia*, pp. 25-34, Oct. 19-24, 2009. Article (CrossRef Link)

[9] S. Ratnasamy, P. Francis, M. Handley, R. Karp, S. Shenker, "A Scalable Content-addressable Network," in *Proc. of the 2001 Conference on Applications, Technologies, Architectures, and Protocol for Computer Communications (Sigcomm)*, vol. 31, no. 4, pp. 161-172, Aug. 27-31, 2001. Article (CrossRef Link)

[10] S. Ratnasamy, M. Handley, R. Karp, S. Shenker, "Topologically-Aware Overlay Construction and Server Selection," in *Proc. of IEEE International Conference on Computer Communications (Infocom)*, pp. 1190-1199, June 23-27, 2002. Article (CrossRef Link)

[11] A. Kermarrec, M. Steen, "Gossiping in Distributed Systems," *ACM Operating Systems review – Gossip-based Computer Networking*, vol. 41, no. 5, pp. 2-7, Oct. 2007. Article (CrossRef Link)

[12] OverSim, The Overlay Simulation Framework, http://www.oversim.org/.

[13] OMNeT++, http://www.omnetpp.org/.

[14] Gnuplot, http://www.gnuplot.info/.

[15] M. Dick, E. Pacitti, B. Kemme, "Flower-CDN: A Hybrid P2P Overlay for Efficient Query Processing in CDN," in *Proc. of 12th International Conference on Extending Database Technology: Advances in Database Technology*, pp. 427-438, Mar. 2009. Article (CrossRef Link)

[16] M. El Dick, E. Pacitti, B. Kemme, "A Highly Robust P2P-CDN under Large-scale and Dynamic Participation," in *Proc. of the First International Conference on Advances in P2P Systems*, pp. 180-185, 2009. Article (CrossRef Link)

[17] X. Liu, H. Yin, C. Lin, "A Novel and High-quality Measurement Study of Commercial CDN-P2P Live Streaming," in *Proc. of the 2009 WRI International Conference on Communications and Mobile Computing*, pp. 325-329, 2009. Article (CrossRef Link)

[18] S. Seyyedi, B. Akbari, "Hybrid CDN-P2P Architecture for Live Video Streaming: Comparative Study of Connected and Unconnected Meshes," in *Proc. of International Symposium on Computer Networks and Distributed Systems*, pp. 175-180, Feb. 23-24, 2011. Article (CrossRef Link)

Kim Thinh Nguyen received the B.S. degree in information technology from Post and Telecommunication Institute of Technology (PTIT), HoChiMinh city, Vietnam, in 2009. He is studying master course in Data Communication Network (DCN) laboratory, Soongsil University, Korea. His research interests include wired and wireless networking, peer-to-peer overlay network.

Young Han Kim received the B. S. degree in electronics engineering from the Seoul National University, Seoul, Korea, in 1984, and the M. S. and Ph. D. degrees in electrical engineering from the Korea Advanced Institute of Science and Technology, Seoul, in 1986 and 1990, respectively. From 1987 to 1994 he was a senior member of the research staff in the Digicom Institute of Telemetics, Seoul, Korea, where he did research on data and computer communication systems. Since September 1994 he has been with Soongsil University, where he is a professor of School of electronics. His current research interests include Wireless Networking Technologies including mobility management, QoS control, Ad-hoc routing, and session control.