Multimedia Tools and Applications (2022) 81:23051–23090
https://doi.org/10.1007/s11042-021-10985-y

1201: VIDEO ON DEMAND OVER OVER THE TOP PLATFORM

Location-proximity-based clustering method for peer-to-peer multimedia streaming services with multiple sources

Changkyu Lee[1,2] · Shin-Gak Kang[1,2] · Anand Nayyar[3]

Received: 16 August 2020 / Revised: 31 January 2021 / Accepted: 30 April 2021 /
Published online: 19 May 2021

Abstract

Social distancing to reduce the spread of coronavirus disease 2019 (COVID-19) made a huge increase in the global OTT market, and OTT service providers get millions of new subscribers. Recently OTT service providers are extending their service to video broadcasting. As a one type of video broadcasting, this paper covers multimedia streaming with multiple sources. Multimedia streaming with multiple sources has multiple sources, and receivers can select one specific source to watch the video from the source. Sources include cameras capturing different angles of same event or location, cameras in geographical locations, etc. For delivering video to rapidly increasing number of users, multimedia streaming with multiple sources system needs efficient and scalable delivery method. Tree-based Peer-to-peer (P2P) networking has been investigated as the delivery solution of multimedia streaming with multiple sources, and set-top boxes or mobile apps of OTT service can be used as peers connecting the subscriber of OTT service. However, the scalability of the tree-based P2P networking is limited by the out-degree of a tree that branches linearly with the number of users. Hence, this study proposes clustering peers based on the location proximity of the peers to enhance the scalability of the P2P multimedia streaming with multiple sources. By clustering peers, one or more peers can be grouped into a virtual peer with an aggregated uplink/downlink capacity. This paper describes P2P multimedia streaming with multiple sources and algorithms for the proposed clustering method. Two applications which are one-view multiparty video conferencing and multi-view video streaming are introduced, and considerations for applying the proposed method to the applications are also discussed. The experimental results show that location-proximity-based clustering is effective in achieving a scalable P2P multimedia streaming with multiple sources by reducing the out-degree of a tree for the introduced applications. The proposed clustering leads improvement in the maximum achievable video bit rate, the average viewing video bit rate, and perceived delay.

✉ Changkyu Lee
changkyu.lee@etri.re.kr

Extended author information available on the last page of the article

Keywords Peer to peer networking · Video conferencing · Multi-view video streaming · Clustering

1 Introduction

Over the Top (OTT) media service offers multimedia streaming over the Internet. Social distancing to reduce the spread of coronavirus disease 2019 (COVID-19) made a huge increase in the global OTT market, and OTT service providers get millions of new subscribers. In addition to Video on Demand (VoD) service, which is the major service of OTT service provider, recently OTT service providers are extending their service to video broadcasting. Among the various types of video broadcasting, this paper covers multimedia streaming with multiple sources. Multimedia streaming with multiple sources has multiple sources, and receivers can select one specific source to watch the video from the source. Sources include cameras capturing different angles of same event or location, cameras in geographical locations, etc. For delivering video to rapidly increasing number of users, multimedia streaming with multiple sources system needs efficient and scalable delivery method.

The simplest delivery method is a client-server approach. Here a user sends his/her audio/video file to a dedicated server which is managed by a service provider. Then, the server relays the received audio/video to the other participants. Evidently, the dedicated server will suffer from relay burden with an increasing number of participants, resulting to low scalability. This low scalability of the client-server approach can be addressed using a content delivery network (CDN) service. The edge servers of the CDN are distributed geographically, and a user can retrieve the content from a nearby server instead of the dedicated server at the service provider side. Thus, the burden on the dedicated server at the service provider side is independent of the number of participants. In addition, by adapting technologies for live streaming over HTTP (e.g., HTTP live streaming [9] and dynamic adaptive streaming over HTTP [10]), the latest CDN service supports live streaming. However, using CDN service is costly, which is closely associated with the volume of outbound traffic. Peer-to-peer (P2P) networking is a cost-effective alternative. It is a communication among nodes, referred to as peers, which are equally privileged as a server and a client. To achieve a certain purpose, such as file sharing and multimedia streaming, peers organize a P2P network, which is a logical network over a physical underlying network. A peer can send and receive data to/from other peers in the P2P network directly instead of depending on a dedicated server. Set-top boxes or mobile apps of OTT media service can be used as peers and organize a P2P network connecting the subscribers of OTT service. Rapid growth of subscription incurs delivery burden on OTT service providers, and so P2P networking can be cost-effective delivery solution. P2P networking has been used as a delivery solution for multimedia streaming with multiple sources such as video conferencing [22, 24, 26]. Y. Xu et al. [26] found that a multimedia streaming with multiple sources for all-view video conferencing can use P2P networking to transmit audios, whereas videos are relayed through dedicated servers. The design relies on the fact that, in the all-view video conferencing, all participants watch each other's video at high quality.[1] However, each one may not have sufficient uplink capacity to send his/her video to all other participants or downlink capacity to receive videos from all of them. On the basis of the hybrid peer-assisted solution [16], P2P multimedia streaming with multiple sources for one-view video conferencing [7, 28] has been investigated as another approach.

[1] A participant can produce a high-quality video and send it. The amount of both the aggregated upload and aggregated download bandwidths increase with the number of participants.

In the P2P multimedia streaming with multiple sources for one-view video conferencing, a participant watches another participant's video in high quality and the rest of the videos in low quality.[2] Thus, the P2P multimedia streaming with multiple sources for one-view video conferencing requires lesser uplink/ downlink capacity than the P2P multimedia streaming with multiple sources for all-view video conferencing. Another important advantage of P2P multimedia streaming with multiple sources for one-view video conferencing [7, 28] is low delay which every multimedia streaming service needs. However, M. Chen et al. [7] reported that the tree-based P2P multimedia streaming with multiple sources [7, 28] suffers from low scalability caused by the out-degree of a tree branching linearly with the number of participants. The limitation also affects the multimedia streaming with multiple sources for other types of service.

To achieve the scalability of the P2P multimedia streaming with multiple sources, this paper proposes clustering peers based on the location proximity of peers. Through peer clustering, one or more peers can be grouped into a single virtual peer with an aggregated uplink/ downlink capacity. Then, the P2P network comprises virtual peers instead of actual peers. The out-degree of a tree can be reduced because the number of virtual peers is not less than that of all of the peers participating in the P2P network. Consequently, the scalability of the P2P multimedia streaming with multiple sources can be improved. The three major contributions of this study are as follows.

1) We propose a method to cluster peers based on their location proximity.
2) We present an analysis on the maximum achievable bit rate.
3) We introduce two applications of P2P multimedia streaming with multiple sources and discuss considerations for applying the clustering method to the applications.
4) We perform experiments to demonstrate the effectiveness of the proposed method in the introduced two applications.

The rest of this paper is organized as follows. Section 2 summarizes related works. Section 3 introduces peer-clustered P2P multimedia streaming with multiple sources, then briefly describes the tree-based P2P multimedia streaming with multiple sources, and fully presents the details of the location-proximity-based clustering method. Section 4 offers an analysis on the maximum achievable video bit rate, and Section 5 introduces two applications of multimedia streaming with multiple sources and discusses considerations for applying the proposed method to the applications. Section 6 discusses the experimental results and findings. Finally, Section 7 elaborates the conclusion of this study.

2 Related work

J. Li et al. [16] proposed Mutualcast for an efficient one-to-many content distribution. In Mutualcast, a source divides its content into fragments. Each fragment is assigned to a particular substream [27]; hence, a peer must receive all substreams to access a content. The source sends each substream to a distinct peer, and the, each peer relays the received substream to the other peers. To minimize the delay in relaying the substream, the

[2] This paper excludes the required upload bandwidth for delivering voice and low-quality video to focus on the delivery of high-quality videos, which require a remarkably large upload bandwidth [16, 28]

maximum number of hops for the relay is limited to two hops. To achieve a high-quality video, the authors proposed utilizing the uplink capacity of the peers, referred to as helpers, who do not request the content. They demonstrated that a P2P network organized into a two-hop tree topology can better perform a one-to-many content delivery.

Using Mutualcast [16], Y. Zhao et al. [28] proposed P2P multimedia streaming with multiple sources for one-view video conferencing. Here, the capacity of the P2P multimedia streaming with multiple sources for one-view video conferencing is calculated on the basis of both the homogeneous environment, where all peers have the same amount of uplink capacity, and heterogeneous environment, where each peer may possess a different amount of uplink capacity. On the basis of the calculated capacity, Y. Zhao et al. [28] derived three guidelines in allocating the bandwidth of peers and also proposed bandwidth allocation algorithms, which are used in distributing multimedia data along the three delivery routes. The experimental results demonstrated that P2P networking can be utilized as an efficient content delivery method for the multimedia streaming with multiple sources for one-view video conferencing. However, M. Chen et al. [7] indicated that content delivery based on a two-hop tree topology is unscalable. The scalability of the P2P multimedia streaming with multiple sources for one-view video conferencing [28] is only limited by the out-degree of the tree, which branches linearly with the number of peers because a peer must relay the received substreams to all other peers. To improve the scalability of the P2P multimedia streaming with multiple sources for one-view video conferencing, we proposed to reduce the number of nodes in the tree by clustering the peers based on their location proximity.

To address user heterogeneity in [28], E. Kurdoglu et al. [15] adopted layered coding and partitioned simulcasting approaches in P2P multimedia streaming with multiple sources. In the layered coding approach, a source generates layered videos to be partially decoded. Then, viewers can select a number of layers according to their capacity and obtain a high-quality video by downloading a large number of layers. Meanwhile, in the partitioned simulcasting approach, a source generates multiple videos with various bit rates, and it sends an appropriate video to the viewers based on their download capacities. According to a numerical comparison, the partitioned simulcasting approach exhibited better performance with regard to average receiving quality and overhead. Thus, the proposed two approaches are effective in addressing user heterogeneity from a video bit rate perspective, but the scalability should also be considered. As M. Chen et al. [7] stated, the scalability of the P2P multimedia streaming with multiple sources for one-view video conferencing [28] is limited by the out-degree of the tree. However, the out-degree of the tree is not considered in the proposed two approaches. Thus, the clustering method proposed in this study is effective in reducing the out-degree of the tree and is applicable to the proposed approaches.

İ. E. AkkuŞ et al. [1] also proposed the utilization of a layered video coding in P2P multimedia streaming with multiple sources for one-view video conferencing. Unlike in [15, 28], peers organize a P2P network in chain topology, and the number of maximum hops for relaying a video is unlimited. When distributing own videos, a source generates a base layer video and an enhancement layer video. Upon receiving a request from other peers, the source sends either the base layer video only or both, but it sends both layers only if it does not relay any other source's video. To increase the number of peers receiving both layers, optimistic heuristics are proposed. Accordingly, the length of a chain where a peer is the head of the chain is considered when a peer joins another chain. For example, for peer i with chain length two and peer j with chain length five, peer j will be moved to the end of the chain in where peers i and j attempt to join, and peer i will be an intermediate node for the relay in the chain. Then, instead of two peers in peer i's chain, five peers receiving peer j's video can receive both layers. As discussed previously, the chain-based P2P multimedia streaming with multiple sources for one-view video conferencing [1] does not limit the maximum number of hops for the relay and thus has no scalability issue. However, having no limit on the chain length may incur a severe delay introduced by the relay, affecting the interactivity of multimedia streaming with multiple sources. The clustering method proposed in this study applied the two-hop limitation similar to the suggestion of J. Li et al. [16] proposed in order to minimize the delay incurred by the relay. In addition, this work describes the consideration on reducing the delay to the target upper bound.

Rec. ITU-T X.603.2 [12] described multipoint-to-multipoint group communication, such as multimedia streaming with multiple sources, and then adopted a popular solution for group communication, referred to as IP multicast. Generally, the IP multicast is not completely deployed over the Internet, but it is supported within a local area network (LAN). On the basis of this insight, ITU-T X.603.2 defined a dedicated node, which is known as the head multicast agent (HMA). The peers in the LAN elect one HMA. Then, the HMAs organize a P2P network in tree topology. Upon receiving data from a source, which is a root node in the tree topology, or a parent HMA, the HMA relays the data to its child HMAs through a unicast. For the non-HMA nodes in a LAN, the HMA also relays the received data through a multicast. However, the drawback of this is evident. Specifically, the scalability and performance are limited by the performance of the HMAs because the uplink capacity of the non-HMA node is not utilized. As an efficient communication method, the approach in this study also applied the IP multicast. For stability and efficiency improvement, the proposed clustering method utilizes all available nodes instead of relying only on one dedicated node in a LAN.

Similar to [12], Z. An et al. [2] proposed a combination of a P2P network and local networks for an audio-video conferencing system. The proposed system had several proxy hosts, of which each LAN has one host, and the proxy hosts organized a P2P network in a binary tree topology. Every participant of a conference sent multimedia data to a conference server, and then, the conference server sent multimedia data to its child nodes in the organized P2P network. Upon receiving the multimedia data from the conference server or a parent node, a proxy host relayed the data to the participants in a LAN through a multicast. The drawback of [2] is that the system relies on the stability of the proxy hosts. In addition, the binary tree topology may encounter a high delay from the conference server to the proxy hosts at the leaf of a tree as the number of LANs increases. To minimize the delay acquired by the relay, we applied a two-hop limitation similar to that proposed by J. Li et al. [16]. In addition, the clustering method proposed in this study utilized all available nodes; thus, the drawback of [2] can be mitigated. X. Tu et al. [23] and X. X. Chen et al. [6] investigated a locality-aware multimedia live streaming service over a P2P network.

Similar to the previously discussed method, their approaches considered the location proximity of peers as a factor in establishing a P2P network. In [23], each peer utilized a physical network model to select a physically closed node as its parent or child. Consequently, the physically closed peers organized a P2P network in tree topology. However, the proposed method is challenging to implement in an actual environment generally because Internet service providers do not share the information regarding the physical underlying network.

Moreover, the proposed method does not involve the consideration on the scalability of the P2P network. The scalability of the P2P network proposed in [23] is limited owing to the number of peers. The proposed method in this work can eliminate the connection between the scalability of the P2P network and the number of peers by clustering peers. Meanwhile, a directory server in [6] provides the list of peers recommended for establishing connections when a peer attempts to join a P2P network. To establish this list, the directory server selects the peers based on buffer status, maximum number of connections, closeness, and quality assessment. With regard to closeness, peers in the same LAN will be the excellent candidate. However, peers in the same LAN will establish multiple unicast connections, which obtain a large amount of traffic in the LAN. Instead of establishing multiple unicast connections, we proposed the use of IP multicast in a LAN.

Instead of P2P multimedia streaming with multiple sources based on tree or chain topology, W. Wu et al. [25] proposed CoolConferencing for P2P multimedia streaming with multiple sources for any-view video conferencing based on mesh topology which does not maintain global structure. With the sight that a conferencing session generally includes less than 15 participants, peers in a session organizes a full-mesh network for exchanging the information such as the neighbor buffermap and delay map. Then, on the basis of the information, the peers can form a mesh network for data delivery. It is well-known that the data delivery over P2P network in mesh topology requires exchanging three messages for pulling data which introduces three one-way delays between two peers, but CoolConferencing reduced the delay by pushing data instead of pulling data. In forming the mesh network for data delivery, a peer selects a dedicated peer, called supplier, and then the supplier will receive the data and push it to the peer. In addition, a peer can receive the data from a helper similar to [16, 28]. However, CoolConferencing highly relies on the suppliers for data delivery, because the suppliers are used like parent nodes in a tree topology. Reducing the burden on the suppliers may increase the performance of a P2P multimedia streaming with multiple sources system. By applying the clustering method, the number of receivers of each supplier can be reduced, and thus the performance of CoolConferencing will be improved.

As an efficient content delivery method over a P2P network, [17, 18, 21] also assessed peer clustering. Peers can be clustered based on various criteria, such as proximity and common-interest. By clustering peers, a P2P network acquires distinct characteristics. For example, the data frequently requested by peers can be replicated into a certain peer in the cluster, resulting in peers finding the desired data from a nearby peer. Further, clustering peers can also be adopted in localizing traffic to reduce the inter-domain traffic. However, the clustering method for improving the scalability of the P2P multimedia streaming with multiple sources has not been investigated yet.

3 Clustered P2P multimedia streaming with multiple sources

3.1 Overview of the P2P multimedia streaming with multiple sources

This Section briefly introduces the tree-based P2P multimedia streaming with multiple sources [28]. Table 1 lists the notations and their corresponding definitions. The P2P multimedia streaming with multiple sources comprises multiple subgroups, and each subgroup is hosted by a video source, and the set of sources generating a video is denoted as *S*. Each subgroup has

Table 1 Notations and definitions

Notations	Definitions
N	Set of peers in a conferencing system
S	Set of video sources
I	Set of idle viewers (No video source)
H	Set of helpers
$H_i,\ i\in S$	Set of helpers of i’s video, $i\in S$
$H_i^R,\ i\in S$	Set of helpers of i’s video with the uplink capacity for delay requirement, $i\in S$
$G_i,\ i\in S$	Set of viewers of i’s video, $i\in S$
$G_i^R,\ i\in S$	Set of viewers of i’s video with the uplink capacity for delay requirement, $i\in S$
$VP_i,\ i\in S$	Set of virtual peers involving subgroup hosted by source i
P_V	Set of peers in a virtual peer
P_H	Peer with the highest ID of a peer among peers of a virtual peer
$r_i,\ i\in S$	Bit rate of video generated by source peer i
$r_i^*,\ i\in S$	Maximum achievable bit rate of video generated by source peer i
$u_i,\ i\in S$	Total upload capacity of peer i
$u_i^s,\ i\in S$	Uplink capacity of a source peer i allocated to the subgroup it is hosting
$u_i^v,\ i\in S$	Uplink capacity of peer i allocated to the subgroup it is watching
$u_i^h,\ i\in S$	Uplink capacity of peer i allocated to the helper pool
u_R^v	Uplink capacity of viewers required to guarantee the target delay
u_R^h	Uplink capacity of helpers required to guarantee the target delay
d_T	Target delay upper bound
S_F	Size of a single fragment of a content
T_{LI}	Predefined timer for location identification
$b_{i,j}$	Bit rate of substream sent by peer i to peer j
$B_i^V,\ i\in S$	Bandwidth contributed for i’s video by viewers
$B_i^{VR},\ i\in S$	Bandwidth contributed for i’s video by viewers belonging to G_i^R
$B_i^H,\ i\in S$	Bandwidth contributed for i’s video by helpers
$B_i^{HR},\ i\in S$	Bandwidth contributed for i’s video by helpers belonging to H_i^R
ID_{PEER}	ID of a peer
ID_{VPEER}	ID of a virtual peer
ID_{LAN}	ID used as a candidate for ID_{VPEER}
ID_{NET}	ID of a P2P network
MSG_{LI}	Message for location identification
MSG_{LA}	Message for location advertisement
MSG_{II}	Message for identification interruption
MSG_{LR}	Message for location report
MSG_{RR}	Message for report response
SVR_M	Management server

a corresponding P2P network organized into two-hop tree topology. At least two subgroups are available in the system considering that every participant watches one high-quality video of another participant.

A set of viewers watching the video generated by sources is also present, and a set of viewers watching the video of source s is denoted by G_s. Evidently, $\sum_{s\in S}|G_s| = |N|$, where N is a set of peers in the conferencing system, as listed in Table 1. Considering that any peer can be a source, viewer i in G_s can be either an idle viewer $i\in I$ or a busy viewer $i\in S$.

Apart from the role of a source and viewer, a viewer can be a helper $k\in H$ by contributing his/her available uplink capacity to a helper pool H to relay a video that he/she is not watching. Any subgroup can borrow the uplink capacity from the helper pool H when necessary, and a set of helpers borrowed by a subgroup hosted by source s is denoted by H_s. Noticeably, $\sum_{s \in S}|H_s| = |H|$. Consequently, peer i with the uplink capacity u_i allocates his/her uplink capacity according to peer i's role; hence, $u_i = u_i^s + u_i^v + u_i^h$, where s, v, and h represent the source, viewer, and helper, respectively. To offer a video of source i at the maximum achievable bit rate, a management server (e.g., peer activity management server [13]) is assumed for setting the maximum achievable video bit rate based on the uplink capacity of a source u_i^s, uplink capacity contributed by all viewers $\sum_{j \in G_i} u_j^v$, the uplink capacity borrowed from a helper pool $\sum_{j \in H_i} u_j^h$. The assumption on the management server is practical, because P2P networking has generally used a management server such as tracker [5]. Use of the management server implies that peers and the management server need to interact for clustering peers. Thus, the management server may not respond quickly when peers join and leave frequently. However, peer churn depends on the type of service. For example, participants will not happen frequently in multiparty video conferencing (MPVC). Participants usually join the conferencing when the conferencing starts and stay until the conferencing is closed. Consequently, use of the management server will not affect the stability of MPVC. For other types of service, we assumed that P2P networks are stable to focus on the effectiveness of the proposed clustering method. This study also assumes that the maximum achievable video bit rate is limited only by the uplink capacity of peers. This assumption is generally adopted by studies on P2P networking [7, 14, 16, 28].

To distribute a video, source s divides its video into multiple substreams [27]. Then each substream can be delivered through three routes as follows:

- Source s sends substreams to viewers; each viewer may receive a different number of substreams. Then, the viewers relay the received substream to $|G_s| - 1$ viewers.
- If a specific target bit rate is not achievable, then source s borrows helpers from a helper pool H and sends substreams to helpers in H_s. Then, the helpers relay the substreams to $|G_s|$ viewers.
- If source s still has available uplink capacity after sending substreams based on the previous two routes, then it can directly send the same substreams to $|G_s|$ viewers to achieve the maximum video bit rate.

3.2 Location-proximity-based clustering

The low scalability of the two-hop tree based P2P multimedia streaming with multiple sources [28] is caused by the out-degree of the tree branching linearly with the number of viewers [7], because the maximum number of hops for the relay is only two hops to minimize the delay introduced during the relay. As a viewer, a peer is responsible for relaying the substreams from source i to $|G_s| - 1$ viewers. As a helper, a peer may also relay the substreams of source j to $|G_j|$ viewers. As a source, peer i sends its substream to $|G_i|$ viewers and it may require to additionally send $|H_i|$ helpers. Therefore, the responsibility of a peer becomes high as the number of viewers increases.

To decouple the out-degree of a tree from the number of viewers, this work proposes to cluster peers based on their location proximity. Through peer clustering, one or more peers form a logical node, referred to as a virtual peer. A virtual peer is a set of peers joining a P2P network from a LAN. For example, many peers reside in a LAN and are the viewers of source

i. Thus, these peers are members of a P2P network organized to deliver the video of source i, and they form a virtual peer. Figure 1 shows an example of a P2P network with virtual peers. Peers B-E are viewers, and peer F is a helper, whereas the root of the P2P network is peer A. Figure 1 assumes that viewers B and C belong to one LAN, whereas viewers D and E are part of another LAN. Through peer clustering peers, peers A and B form a virtual peer, and peers D and E form another virtual peer.

After the peers are clustered, the P2P network comprises virtual peers instead of realistic ones. Specifically, clustering peers shifts viewer set G_s into virtual peer set VP_s. Accordingly, the delivery route of the substream is affected. A viewer relays substreams to $|VP_s| - 1$ virtual peers instead of $|G_s| - 1$ peers to deliver a video from source s. Subsequently, a helper relays substreams to $|VP_s|$ virtual peers instead of $|G_s|$ peers, whereas source s sends substreams to $|VP_s|$ peers instead of $|G_s|$ peers as the third delivery route. That is, substreams are relayed to one peer for each virtual peer. Hence, the out-degree can be reduced when at least two peers form a virtual peer. To efficiently maximize the uplink capacity of peers, source s sends substreams to all available viewers in G_s as the first delivery route and then to all helpers in H_s as the second delivery route.

The peers of a virtual peer must share the received substreams considering that each substream is relayed to one peer for each virtual peer. Generally, a peer can completely utilize its uplink capacity within a LAN, whereas the communication over the Internet is limited by bottlenecks. For example, the uplink capacity within a LAN can reach 100 Mbps, 1 Gbps, or even higher. Assuming 1 Mbps substreams, which is sufficient, the substreams can be shared within 10 ms in a 100 Mbps LAN and 1 ms in a 1 Gbps LAN, which are regarded as a negligible delay. However, a unicast transmission among peers of a virtual peer also augments the out-degree, directly affecting the scalability of the tree-based P2P multimedia streaming with multiple sources. To achieve a scalable P2P multimedia streaming with multiple sources, an IP multicast, which is an efficient group communication method, is adopted to shared substreams among peers in a virtual peer. If the IP multicast is not supported in a specific LAN, then each peer in the LAN becomes a single virtual peer. After clustering, the peers of a virtual peer can be regarded as a single virtual peer with an uplink capacity as $\sum_{j \in VP_i} u_j$, where peer j allocates u_j as the uplink capacity of a unicast communication. Apart from sharing substreams, peers also

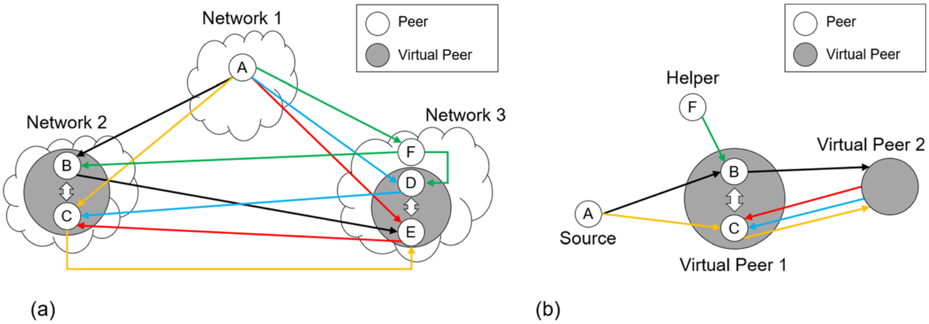


Fig. 1 **a** Example of a P2P network, where peer A is the source, peer F is the helper, and peers B-E are viewers. Arrows indicates the transmission of substreams: the white arrow denotes a multicast transmission, whereas the other arrows indicate a unicast transmission. **b** Virtual peer 1's perspective of (a)

use an IP multicast in identifying location proximity that peers execute, thus forming a virtual peer.

3.3 Location identification for location-proximity-based clustering

This Section describes the method used in identifying the location of peers. All peers periodically conduct location identification, and this operation is important for the proposed method. The first reason is that peers use location information to determine their proximity with other peers, and then, the peers in a LAN can form a virtual peer. Second, every peer should know the location of other peers to transmit substreams accurately. For example, in Fig. 1, peer B requires the location information of other peers (viewers), which are peers C-E. Otherwise, peer B relays substreams to three peers through unicast, which unnecessarily increases the responsibility of peer B and the out-degree of a tree. In identifying peers, this study assumes that each peer has a unique ID ID_{PEER} and that each P2P network can be identified through its unique ID ID_{NET}. These IDs can be generated using various methods, e.g., SHA-1 [19]. These methods are commonly used in applications via P2P networking, such as BitTorrent [5], which is a popular application for sharing files. In addition, every virtual peer is also assumed to possess a unique identifier ID_{VPEER}. However, the method of generating the unique IDs is not the scope of this study.

In Algorithm 1, we present the pseudocode of the location identification that each peer executes. First, a peer generates a random ID representing the LAN where he belongs, which is denoted by ID_{LAN}. Then the peer starts the location identification by multicasting a location identification message MSG_{LI}, including ID_{PEER}, ID_{NET}, and ID_{LAN}. Here, the ID_{NET} in the MSG_{LI} represents the P2P network where the peer sending the message is participating. A peer in a LAN will receive at least one MSG_{LI} sent by another peer, which is participating in the same subgroup, when two or more peers reside in the LAN. Upon receiving the MSG_{LI}, a peer verifies whether the peer who sent the message belongs to the same P2P network by referring to the ID_{NET} indicated in the received message. If these peers are in the same P2P network, then the peer compares its own ID_{PEER} with the ID_{PEER} of the received MSG_{LI}. The ID_{LAN} generated by the peer with the highest ID_{PEER}, denoted by P_H, is selected as the ID_{VPEER} of the virtual peer. For example, four peers are present, peers A-D, in a LAN. Peers A and B belong to a P2P network, whereas peers C and D belong to another P2P network. Thus, peers A and B form a virtual peer, whereas peers C and D form another virtual peer. If ID_{PEER} follows an alphabetical order of each peer, then, the ID_{VPEER} of one virtual peer will be the ID_{LAN} generated by peer B. Another virtual peer will have an identifier, that is, ID_{LAN}, generated by peer D. After a peer sets itself as P_H, the peer periodically multicasts an identification interruption message, denoted by MSG_{II}, which including ID_{PEER}, ID_{NET}, and ID_{VPEER}. Upon receiving the message, other peers in the LAN postpone the location identification operation until they receive a location advertisement message. A peer joining after the previously mentioned operations will receive MSG_{II} and then it also postpones the location identification operation.

A peer will receive a location advertisement message, which including ID_{PEER}, ID_{NET}, and ID_{VPEER}, sent by P_H,if P_H has been elected before the peer joins. Upon receiving the location advertisement message, denoted by MSG_{LA}, the peer directly sets the peer who sent the MSG_{LA} and ID_{VPEER} in the received MSG_{LA} as P_H and ID_{VPEER}, respectively. A peer can be aware that no other peer is present in its LAN if neither MSG_{LI}, MSG_{II} nor MSG_{LA} received prior to a predefined time, denoted by T_{LI}, is expired. In such case, the peer sets itself and its own ID_{LAN} as P_H and ID_{VPEER}, respectively.

Algorithm 1 Location identification on the peer side

```
procedure locationIdentificationPeer($ID_{PEER}$, $ID_{NET}$, $ID_{LAN}$)
    for each peer $j \in N$ do
        multicast a location identification message ($MSG_{LI}$) including $ID_{PEER}$, $ID_{NET}$, and $ID_{LAN}$
        while $T_{LI}$ is not expired ▷start of $P_H$ election
            if $j$ receives $MSG_{LI}$ sent by peer $k$ then
                select the peer with higher $ID_{PEER}$ as $P_H$
                set $ID_{LAN}$ generated by $P_H$ as $ID_{VPEER}$
            else if $j$ receives an identification interruption message ($MSG_{II}$) or
                    a location advertisement message ($MSG_{LA}$) sent by peer $k$ then
                set $k$ as $P_H$
                set $ID_{VPEER}$ sent by $k$ as $ID_{VPEER}$
            end if
        end while
        if $P_H$ is not set then
            set $j$ as $P_H$
            initiate periodical interruption
            set own $ID_{LAN}$ as $ID_{VPEER}$
        end if ▷ end of $P_H$ election
        if $j$ is $P_H$ ▷ start of location report then
            send a location report message ($MSG_{LR}$) to a management server ($SVR_M$)
            if $j$ receives a report response message ($MSG_{RR}$) then
                if $ID_{VPEER}$ is duplicated then
                    change $ID_{VPEER}$ as proposed by $SVR_M$
                end if
                initiate periodical location advertisement
                terminate periodical interruption
                initiate periodical location report
            end if
        else
            if $j$ receives $MSG_{LA}$ then
                initiate periodical location report
            end if
        end if ▷end of location report
    end for
end procedure
```

After electing P_H, it sends the management server (SVR_M) a location report message (MSG_{LR}), including ID_{PEER}, ID_{NET}, and ID_{VPEER}. P_H changes its ID_{VPEER} based on the response message, denoted by MSG_{RR}, from SVR_M when MSG_{RR} shows the duplication of ID_{VPEER} and has alternative ID_{VPEER}. After checking ID_{VPEER}, P_H multicasts MSG_{LA} periodically. The MSG_{LA} commands the non-P_H peers to report their location to the SVR_M and also prevents a newly joining peer in the P2P network from executing unnecessary location identification. Note that the ID_{VPEER} of a virtual peer will not change until all of the peers in a LAN leave the corresponding P2P network even if the newly joined peer has ID_{PEER} higher than that of P_H. Moreover, P_H sends MSG_{LR} to SVR_M periodically. Upon receiving MSG_{LA}, the non-P_H peers begin reporting their locations to SVR_M by sending MSG_{LR} including ID_{PEER}, ID_{NET}, and ID_{VPEER}. On the basis of the report from P_H and non-P_H peers, SVR_M can maintain the location information of all peers. The non-P_H peers execute location identification again when they do not receive MSG_{LA} prior to a predefined time or when the time expired.

Algorithm 2 lists the pseudocode of the location identification executed by the management server (SVR_M). Upon receiving MSG_{LR} from a peer, SVR_M verifies whether ID_{NET} and ID_{PEER} in the received MSG_{LR} are valid. The reported ID_{NET} is considered invalid when a P2P network corresponding to ID_{NET} has not been established. Thus, SVR_M depends on the response message, denoted by MSG_{RR}, indicating an error. However, SVR_M does not address the invalidity of the reported ID_{PEER} as an error because it happens when a peer who sent MSG_{LR} recently joins. Hence, SVR_M must register the newly joined peer on its database. Otherwise, SVR_M updates the already registered information about the peer who sent MSG_{LR}. After verifying two IDs, SVR_M assesses whether the other virtual peer is using ID_{VPEER} in the received MSG_{LR}. If the reported ID_{VPEER} is already used by the other virtual peer, then SVR_M responds with a newly generated ID_{VPEER}. Otherwise, SVR_M stores the reported information and responds with a code indicating confirmation.

Algorithm 2 Location identification on the server side

```
procedure locationIdentificationSvr($MSG_{LR}$)
    if a management server ($SVR_M$) receives a location report message ($MSG_{LR}$) sent by peer j,
      $j \in N$ then
        if $ID_{NET}$ is not valid then
            send a report response message ($MSG_{RR}$) indicating an error
        else
            if $ID_{PEER}$ is valid then ▷for an existing peer
                if no $ID_{VPEER}$ Collision then
                    refresh the stored information
                    respond with $MSG_{RR}$ indicating confirmation
                else
                    generate an alternative ID for $ID_{VPEER}$
                    refresh the stored information
                    respond with $MSG_{RR}$ indicating duplication
                end if
            else ▷for a peer newly joining P2P network
                if no $ID_{VPEER}$ collision then
                    store the reported IDs
                    respond with $MSG_{RR}$ indicating confirmation
                else
                    generate an alternative ID for $ID_{VPEER}$
                    store the reported IDs with the new $ID_{VPEER}$
                    respond with $MSG_{RR}$ indicating duplication
                end if
            end if
        end if
    end if
end procedure
```

No operation is required for peers to form a virtual peer because the location identification implicitly clusters peers. By performing the location identification, peers will be aware of the existence of other peers belonging to the same P2P network. If at least two peers are present in a LAN, they all multicast the received substreams into the LAN. Otherwise, the substreams will not be shared in the LAN until another peer in the LAN joins in this P2P network.

Thus far, this Section has described the identification of peers' location and the organization of virtual peers. To send substreams, peers, including sources, viewers, and helpers must be aware of the location information of the peers in a P2P network, and this location information is provided by SVR_M. As discussed previously, SVR_M maintains the reported information including the ID_{PEER}, ID_{NET}, ID_{VPEER}, ID_{PEER} of P_H in the virtual peer identified by ID_{VPEER}, and the location information of the peer. Note that SVR_M can obtain the location information of each peer when it receives MSG_{LR}. Further, it provides the location information of peers when it responds to MSG_{LR} from peers. On the basis of the role of a peer, SVR_M provides different types of location information as follows:

- To source s: Source s hosts a subgroup corresponding to a P2P network and must send substreams to every viewer in G_s. Thus, SVR_M sends the location information of all peers in its P2P network to source s.
- To P_H: Among the peers in the subgroup of source s, only P_H of each virtual peer receives the location information of peers in different virtual peers. The design primarily aims to minimize the responsibility of SVR_M. P_H shares the information with the non-P_H peers in a virtual peer by sending MSG_{LA}.

On the basis of the location information, all peers can send and relay substreams. It needs to be prevented that one specific peer receives substreams from all other virtual peers and multicasts the substreams into its LAN, because the peer is imposed by much burden in such case. To prevent such an extreme case and to balance the responsibility among peers in a virtual peer, the location information is selectively provided by SVR_M. When SVR_M sends MSG_{RR} with location information to P_H, it does not offer the location information of all of the peers. Alternatively, SVR_M selects a different portion of peers in every virtual peer for each P_H. For example, three virtual peers are present. Virtual peer 1 comprises peers A, B, C, and D. Virtual peer 2 contains peers E, F, and G. Meanwhile, virtual peer 3 is composed of peers H, I, and J. P_H of virtual peer 1 is peer D, that of virtual peer 2 is G, and that of virtual peer 3 is J. Thus, peer G of virtual peer 2 can receive the location information of peers A and B of virtual peer 1, whereas peer J of virtual peer 3 can receive the location information of peers C and D. Consequently, peers in virtual peer 2 can relay substreams to either peer A or B, whereas peers in virtual peer 3 can relay substreams to either peer C or D.

4 Analysis regarding the maximum achievable video bit rate

Reducing the responsibility of peers has two effects. First, clustering peers improves the scalability of a two-hop tree based P2P multimedia streaming with multiple sources. As mentioned previously, the scalability of a P2P multimedia streaming with multiple sources is limited by the out-degree that branches linearly with the number of viewers. Clustering peers can effectively disconnect the out-degree from the number of viewers because one or more peers can be grouped into one virtual peer. This reduced burden on the peers results in the second effect of clustering. Second, clustering peers can improve the maximum achievable video bit rate of each subgroup. As the amount of the required uplink capacity for relaying substreams is reduced via peer clustering, the peers can relay substreams with high bit rate. In this Section, the following three theorems show the maximum achievable video bit rate of the clustered P2P multimedia streaming system with multiple sources. Here, the technique of increasing the bit rate as the number of peers in a virtual peer increases is discussed.

Theorem 1 The location-proximity-based clustering achieves the minimum performance, if $|P_V| = 1$.

Proof We will explain the maximum achievable bit rate of the system with virtual peers and then show that $|P_V| = 1$ generates the minimum performance. Note that we extended the analysis in [28] to derive the maximum achievable bit rate. To share the substreams appropriately, viewers and helpers should relay the substreams as received. As listed in Table 1, $b_{i,j}$ denotes the bit rate of the substreams from peer i to peer j. Source i must allocate the uplink capacity when sending substreams to $|G_i| = 1$ viewers as

$$u_{i,V}^{s} = \sum_{j\in G_i} b_{i,j}, u_i^s \geq u_{i,V}^s \tag{1}$$

Viewer j ∈ G_i allocates his/her uplink capacity as a viewer.

$$u_j^v = b_{i,j} \times (|VP_i| - 1) \tag{2}$$

Then, $b_{i,j}$ of viewer j can be calculated using the following equation:

$$b_{i,j} = \min\left(\frac{u_{i,V}^s}{|G_i|}, \frac{u_j^v}{|VP_i - 1|}\right) \tag{3}$$

Helper $k \in H$ is responsible for relaying the substreams to $|VP_i|$ viewers when source i borrows helpers from the helper pool. Moreover, source i allocates the uplink capacity when sending the substreams to $|H_i|$ helpers as

$$u_{i,H}^{s} = \sum_{k\in H_i} b_{i,k}, u_i^s \geq u_{i,H}^s \tag{4}$$

Helper k allocates his/her uplink capacity as a helper as

$$u_k^h = b_{i,k} \times |VP_i| \tag{5}$$

Here, $b_{i,k}$ of helper k is calculated using the following equation:

$$b_{i,k} = \min\left(\frac{u_{i,H}^s}{|H_i|}, \frac{u_k^h}{|VP_i|}\right), u_i^s = u_{i,V}^s + u_{i,H}^s \tag{6}$$

Considerably, each source sends substreams to viewers and helpers who have sufficient uplink capacity. This is reasonable because peers without adequate uplink capacity cannot properly relay the substreams from each source. We also assumed that every source shares a video at the maximum achievable bit rate that can accommodate all viewers and helpers.

On the basis of (1), (2), and (3), the achievable video bit rate through $|G_i|$ viewers is

$$B_{S1} = \sum\nolimits_{j\in G_i} b_{i,j} = \sum\nolimits_{j\in G_i} \frac{u_j^v}{|VP_i|-1} = \frac{B_i^V}{|VP_i|-1} \tag{7}$$

On the basis of (4), (5) and (6), the achievable video bit rate through $|H_i|$ helpers is

$$B_{S2} = \sum\nolimits_{k\in H_i} b_{i,k} = \sum\nolimits_{k\in H_i} \frac{u_k^h}{|VP_i|} = \frac{B_i^H}{|\mathrm{VP}_i|} \tag{8}$$

As a third delivery route, source i can additionally contribute his/her available uplink capacity. Then, the achievable video bit rate from the third delivery route becomes

$$B_{S3} = \frac{u_i^s - B_{S1} - B_{S2}}{|VP_i|} \tag{9}$$

Given that a video bit rate cannot exceed the uplink capacity of the source, the maximum achievable video bit rate of a subgroup hosted by source i, r_i^*, is

$$r_i^* = \min\left(u_i^s, B_{S1}+, B_{S2}+, B_{S3}\right) \tag{10}$$

On the basis of (10), the maximum achievable video bit rate of a subgroup hosted by source i, r_i^*, is affected by the number of virtual peers, $|VP_i|$. We assume that every virtual peer has one peer only; thus, $|P_V| = 1$. In this case, the number of virtual peers is the same number of viewers in a subgroup hosted by source i. Thus, the maximum achievable video bit rate is the same as that presented in state-of-the-art systems [28]. The result can be intuitively realized because $|P_V| = 1$ indicates that clustering is not applied.

Theorem 2 For a certain subgroup with $|G_i|$ viewers, the location-proximity-based clustering can improve the video bit rate of the subgroup hosted by source i, r_i, as the number of virtual peers, $|VP_i|$, is decreased.

Proof We assume that the number of peers in every virtual peer of the subgroup hosted by source i is equal. Then, the number of viewers of the subgroup, $|G_i|$, can be expressed as follows:

$$|G_i| = |VP_i| \times |P_V| \tag{11}$$

Equations (7) to (10) imply that the maximum achievable video bit rate of the subgroup hosted by source i, r_i^*, is affected by $|VP_i|$, whereas (11) indicates that the number of virtual peers, $|VP_i|$, is reduced as the number of peers in a virtual peer, $|P_V|$, is increased. Consequently, the difference between the maximum achievable video bit rate and uplink capacity of source i is reduced as $|VP_i|$ also decreased.

Assume a different assumption wherein every virtual peer has a random number of peers. In this case, the number of viewers of the subgroup, $|G_i|$, can be expressed as

$$|G_i| = \sum\nolimits_{j\in VP_i} \left|P_V^j\right| \tag{12}$$

, where $\left|P_V^j\right|$ is the number of peers in virtual peer j.

With $|G_i|$ viewers, increasing $|P_V|$ of a virtual peer in the subgroup hosted by source i indicates that the $|P_V|$ of another virtual peer in the same subgroup is reduced. Consider two virtual peers m and n that are located at LAN 1 and LAN 2, respectively. In LAN 1, several peers in virtual peer m can leave this virtual peer when they want to watch video of a different source. Meanwhile, some peers in LAN 2 join virtual peer n to watch the video of source i. In a random $|P_V|$ case, $|VP_i|$ is not always affected by the change of $|P_V|$. The change of $|P_V|$ affects the $|VP_i|$ only when all peers of a certain virtual peer leave the subgroup, indicating that $|P_V|$ of the virtual peer is zero. Then, $|VP_i|$ will be reduced. Any other virtual peer can accommodate the same number of peers to obtain $|G_i|$ viewers. As $|P_V|$ is reduced, the maximum achievable video bit rate can be improved.

Theorem 3 The location-proximity-based clustering will achieve the maximum video bit rate of the subgroup hosted by source i, r_i^*, when the number of peers in a virtual peer is the same as the number of viewers of a subgroup.

Proof When the number of peers in a virtual peer, $|P_V|$, is the same as the number of viewers of the subgroup hosted by source i, $|G_i|$, one virtual peer will be organized. According to (10), the maximum achievable video bit rate is equal to the uplink capacity of source i. The video bit rate is in its maximum because the video bit rate cannot exceed the uplink capacity of the source.

5 Applications of multimedia streaming with multiple sources

This section introduces two different applications of multimedia streaming with multiple sources and also describes considerations on applying the clustering method to each application.

5.1 One-view multiparty video conferencing

Video calling has rapidly gained popularity owing to the spread of network-enabled consumer electronics with a built-in camera. Social distancing to reduce the spread of coronavirus disease 2019 (COVID-19) made a huge increase in video calling including multiparty video conferencing (MPVC). Current calling applications such as Skype [22], Google+ Hangout [8], and WebRTC [24] support multiparty video conferencing. MPVC achieves real-time group communication in which every participant can start a video call to the other participants. In one-view MPVC, a participant watches another participant's video in high quality and the rest of the videos in low quality. Thus, the one-view video conferencing requires lesser uplink/downlink capacity than the all-view MPVC. Figure 2 depicts the conceptual image of one-view multiparty video conferencing.

5.1.1 Consideration on the transmission delay

Section 4 demonstrated that the proposed location-proximity-based clustering method can improve the maximum video bit rate. This Section considers another significant

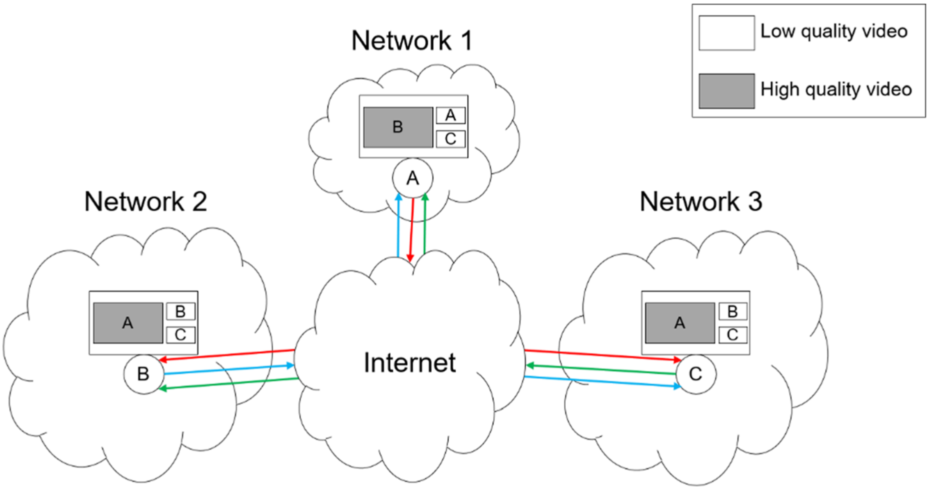


Fig. 2 Conceptual image of one-view multiparty video conferencing

factor apart from the video bit rate, that is, the transmission delay. Transmission delay is one of the important factors that affect the quality of video conferencing. Low delay results in better experience. The delay is closely related to two factors, namely, the amount of uplink capacity allocated for transferring substreams and the size of substream fragments. S_F denotes the size of a fragment, and all fragments are assumed to have equal size. Let us assume that a subgroup has two viewers, i and j, belonging to the same LAN segment and that the available uplink capacity and fragment size are u_i^v = 384 Kbps, u_j^v = 1 Mbps, and S_F = 8 KB (=64 Kb), respectively. Then, the transmission delay will be approximately 300 ms from a source to viewer i and 128 ms from the source to viewer j. When the subgroup includes viewer k in a different LAN segment, the transmission delay from the source to viewer k through viewer i will be approximately 600 ms, and the delay through viewer j will be 256 ms. Moreover, the delay perceived by viewer k will be approximately 600 ms because viewer k requires both fragments. Thus, a specific requirement on the amount of uplink capacity in distributing a fragment within a specific target delay is present. The available uplink capacity of viewers must be equal to or larger than the specific required uplink capacity, which is denoted by u_R^v, guaranteeing the target delay. The required uplink capacity as a viewer of source i is calculated as follows:

$$u_R^v = 2 \times \frac{S_F}{d_T} \times (|VP_i|{-}1) \tag{13}$$

, where d_T denotes the target delay upper bound.

For example, the required uplink capacity of a viewer will be 1.28 Mbps when S_F is 8 KB and d_T is 100 ms. Hence, both viewers i and j will not be selected as the ones who relay the received fragments to another viewer.

To reduce the transmission delay introduced by relaying substreams through helpers, helpers are also required to possess a certain amount of available uplink capacity. The required uplink capacity for a helper of source i, u_R^h, is calculated as follows:

$$u_R^h = 2 \times \frac{S_F}{d_T} \times |VP_i| \tag{14}$$

By considering the requirement for the minimum uplink capacity of each viewer and helper, (7) and (8) are revised to (15) and (16), respectively.

$$B_{S1}^{'} = \sum_{j \in G_i^R} b_{i,j} = \frac{\sum_{j \in G_i^R} u_j^v}{|VP_i| - 1} = \frac{B_i^{VR}}{|VP_i| - 1} \tag{15}$$

, where G_i^R is the group of viewers with the required uplink capacity and B_i^{VR} is the bandwidth contributed for i's video by the viewers belonging to G_i^R.

$$B_{S2}^{'} = \sum_{k \in H_i^R} b_{i,k} = \frac{\sum_{k \in H_i^R} u_k^h}{|VP_i|} = \frac{B_i^{HR}}{|VP_i|} \tag{16}$$

, where H_i^R is the group of helpers with the required uplink capacity and B_i^{HR} is the bandwidth contributed for i's video by the helpers belonging to H_i^R.

Equations (15) and (16) indicate that a certain peer without sufficient uplink capacity will not relay a substream to other viewers in the same subgroup as a viewer or in other subgroups as a helper. Note that a peer will receive substreams from other viewers, but it does not receive any substream directly from a source because it will not relay the substream. Thus, the refined P2P one-view MPVC system may utilize few peers based on the uplink capacity requirement, and the video bit rate offered by the refined system may be lower than that offered by the system as discussed in Section 4.

5.2 Multi-view video streaming

Multi-view video streaming fulfils the delivery of video streams captured simultaneously from multiple camera viewpoints. Users can select their preferred angle of video. Well-known applications of multi-view video streaming include 3D video, free viewpoint TV, and virtual reality (VR). Watching a specific team's view point in a sports game such as football match is another example of multi-view video streaming. Each source sends a video stream of a different angle to the receivers geographically distributed over the Internet. Figure 3 shows the conceptual image of multi-view video streaming.

5.2.1 Consideration on video quality configuration

For video streaming service, it is important to configure the appropriate video bit rate. The Bit rate directly affects the quality of service. In server-based delivery of video stream, such as HLS [9] and DASH [10], service provider can provide the information of mapping between video quality and bit rate. Then, on the basis of the mapping information, each viewer can select a specific quality of video with respect to the network condition such as downlink capacity. However, the achievable video bit rate in the P2P one-view multimedia streaming system considered in this paper is affected by the uplink capacity contributed by viewers. Another important point is that all sources need to provide the same quality of multimedia streams, which are encoded at the same bit rate, to offer the uniform quality of service when a viewer changes view

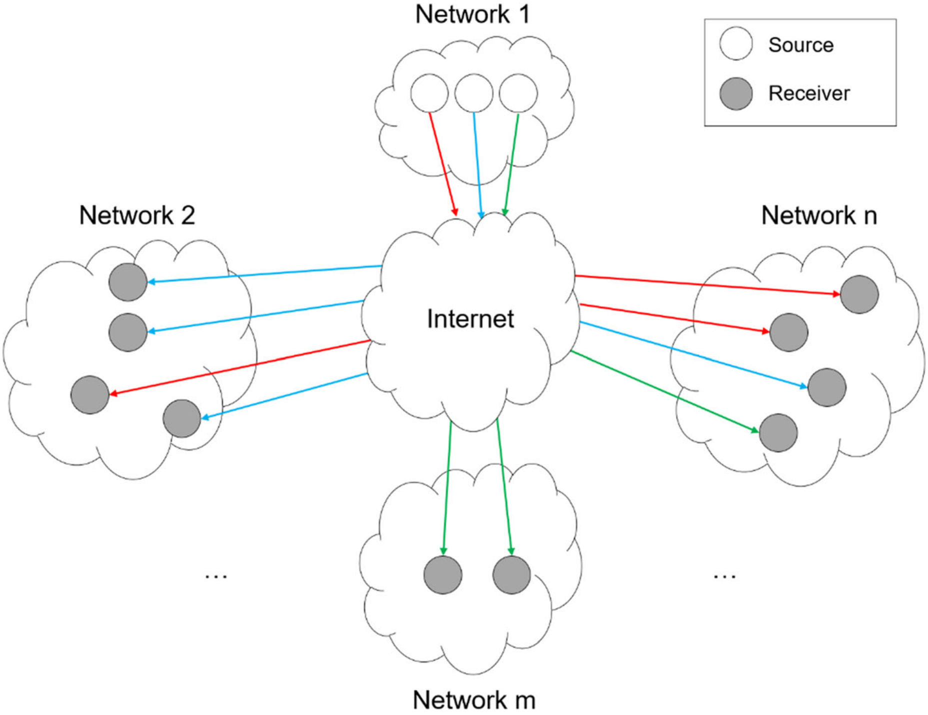


Fig. 3 Conceptual image of multi-view video streaming

point. Thus, configuring a certain bit rate across all P2P networks on the basis on the contribution from sources and viewers is important. This section describes how a management server, which is generally used in modern P2P communications, can configure the bit rate of multimedia streams and that of substreams. We assumed that the management server manages the maximum uplink capacity all peers. To realize the assumption, as described in section 3.3, each peer periodically sends a location report message to the management server after location identification. The message needs to be extended to have the information about the maximum uplink capacity and the available uplink capacity of the reporting peer. It is also assumed that the management server already knows the maximum uplink capacity of sources. The assumption is practical, since every source needs to interact with the management server when they establish a P2P network. When a source requests the management server to generate a P2P network, he/she can notify the management server of his/her maximum uplink capacity. On the basis of the information about uplink capacity, the management server can configure the uplink capacity of viewers and helpers. Based on the reported information, the management server checks whether the multi-view streaming system can support a given target bit rate of multimedia streams and determine the appropriate bit rate across P2P networks as described in Algorithm 3 and Algorithm 4. b_T denotes the bit rate of multimedia streams across all sources, and the value of b_T cannot exceed uplink capacity of each source according to (10). u_i^{sR} and u_j^R denote the available uplink capacity of source i and that of peer j, respectively. H_i^U denotes the set of unallocated helpers which are viewers of source i, and u_i^H denotes the total uplink capacity of unallocated helpers in H_i^U. R_i denotes the required contribution from the second and third routes to distribute multimedia streams of source i.

```
Algorithm 3 Configuring Bit rate for Viewers
procedure bitRateConfViewer($b_T$)
   for each viewer set $G_i$, i ∈ S do
      for each viewer j, j ∈ $G_i$do
         $b_{i,j} = \min(b_T, u_j/(|VP_i| - 1))$
         $u_j^v = b_{i,j} \times (|VP_i| - 1)$
         $u_j^R = u_j - u_j^v$
         $B_{S1} = B_{S1} + b_{i,j}$
         $b_T = b_T - b_{i,j}$
         if $b_T == 0$ then
            set $u_k^R = u_k$ for the rest viewers
          $H_i^U$= available viewers in $G_i$
          break
        end if
     end for
     if $b_T \neq 0$ then
        $R_i = b_T$
        $u_i^{sR} = u_i^s - B_{S1}$
     end if
  end for
  if there is a viewer set $G_i$, $R_i \neq 0$ do
     run Algorithm 4
     if Algorithm 4 returns false then
        go back Line 1 with reduced $b_T$
     end if
  end if
end procedure
```

Algorithm 3 describes configuration of bit rate of substream for each viewer and it interacts with Algorithm 4 to find appropriate bit rate for multimedia streams for all sources. With a certain initial bit rate of multimedia streams, b_T, the bit rate of substream, $b_{i,j}$, for each viewer can be calculated. As the substream bit rate for viewers is configured, the management server also calculates u_j^v of each viewer. If the contribution from viewers is enough to distribute multimedia stream from source i, the management server configures the uplink capacity u_j^R of the rest viewers as u_j so that they can contribute their remaining uplink capacity as a helper. Then the server configures available helper set H_i^U. The helper set includes the viewers that $u_j^R \neq 0$. For each viewer set, the server configures the contribution through the first delivery route, which is B_{S1}, and the bit rate of substream for each viewer. If the contribution from the first route is not enough to distribute multimedia stream, $R_i \neq 0$, the management server configures the second and third routes. Algorithm 4 is used to configure the contribution from the two routes. Before configuring the two routes, the management server calculates how much contribution each source needs to deliver multimedia stream and how much uplink capacity each source has. If Algorithm 4 returns failure, Algorithm 3 starts again with the reduced b_T and this indicates that the target bit rate is not supportable in the system.

```
Algorithm 4 Configuring Bit rate for Helpers
procedure bitRateConfHelper()
  for each source i, $i \in S, R_i \neq 0$ do
    for each set $H_j^U$, $j \in S, j \neq i$ do
      for each helper k, $k \in H_j^U$ do
        $b_{i,k} = \min(R_i, u_k^R/|VP_i|)$
        $u_k^h = b_{i,k} \times |VP_i|$
        $u_k^R = u_k^R - u_k^h$
        $B_{S2} = B_{S2} + b_{i,k}$
        $R_i = R_i - b_{i,k}$
        remove helper k from set $H_i^U$
        if $R_i == 0$ then
          break
        end if
      end for
      if $R_i == 0$ then
        break
      end if
    end for
    $u_i^{sR} = u_i^{sR} - B_{S2}$
    if $R_i \neq 0$ then
      if $R_i > u_i^{sR}$ then
        returns false
      else
        $b_{i,VP_i} = R_i/|\mathrm{VP}_i|$
      end if
    end if
  end for
  return true
end procedure
```

For any source which needs the second and the third routes, the server configures helpers. The management server sets the bit rate of substreams for each helper, $b_{i,\ k}$. When the server configures the bit rate, it is possible to exclude certain helpers which do not have available uplink capacity larger than a certain value. If the contribution from the second route is not enough, $R_i \neq 0$, the management server configures the third route. Through the third route, all virtual peers of source i will receive the substream encoded at b_{i,VP_i}. The configuration will be failed, if the required contribution is greater than the available uplink capacity of source. Then Algorithm 4 returns false, and the system cannot support the target bit rate, b_T. According to Algorithm 3, the management server reduces b_T and runs Algorithm 3 again. The management server runs both algorithms periodically to find the appropriate bit rate of multimedia streams across all sources and that of substream for each viewer and helper. Through the periodic operation, multi-view streaming system can apply the latest status of viewers. The management server notifies all sources of the result after running algorithms. Based on the result, sources can start the multi-view streaming service with the calculated bit rate.

6 Experiment results and findings

The experimental results presented in this Section show the effectiveness of clustering peers for each application introduced in Section 5.

6.1 Effectiveness on P2P one-view multiparty video conferencing

To allocate the uplink capacity of peers and observe the results, C++ code was implemented for numerical evaluation, and 4000 independent runs per experiment were executed. The total number of peers, $|N|$, was varied from 6 to 12, with a step size of 2. The assumption about the number of participants is reasonable, because a conferencing does not have more than 15 participants in general [20]. For heterogeneity, every peer had an uplink capacity randomly set from 128 Kbps to 5 Mbps based on the measurement assessment [4]. Three different aspects of the results were measured. First, the maximum achievable video bit rate was obtained to demonstrate the performance of the system. Second, the average viewing video bit rate was achieved because the maximum achievable video bit rate may not accurately show the video bit rate encountered by each peer. Finally, the sum of the unused uplink capacity of all peers was acquired to show the scalability of the system. A large amount of the unused uplink capacity indicates that the system can accommodate a large number of viewers.

As considered in Theorem 2, the first experiment was performed with a configuration wherein every virtual peer had an equal number of peers. Figure 4 plots the CDF of the maximum achievable video bit rate offered by sources for each $|N|$. The figure also shows that an increase in the $|P_V|$ value, which indicates the number of peers in a virtual peer, increases the number of sources that can offer a video at a higher bit rate.

This result is because each viewer and helper are responsible for relaying the substreams to a few number of viewers as $|P_V|$ increases. Owing to the reduced relay issue, viewers and helpers can contribute additional uplink capacity to relay substreams to ensure each source can increase $b_{i,j}$ for the viewers and helpers who contribute an extra amount of uplink capacity. Consequently, the increased $b_{i,j}$ improves the maximum video bit rate.

Figure 5 plots the CDF of the average video bit rate perceived by all peers. In every scenario, a greater $|P_V|$ results in the higher average viewing video bit rate, which is directly related to the quality of the video. The results clearly show that the clustering peers can improve the quality of the experience for the viewers. In addition, $|S|$ affects the average video bit rate. Considering that each source is watched by at least one viewer, an increasing $|S|$ directly indicates that the number of viewers of a certain source is decreasing. However, the tendency becomes aggressive with high $|P_V|$. As observed in the results depicted in Figs. 4 and 5, we conclude that clustering peers improves the video quality performance of the P2P one-view MPVC.

Figure 6 plots the CDF of the sum of unused uplink capacity of all peers. Note that the aggregated unused uplink capacity is increased with high $|P_V|$.

Given that clustering peers improves video bit rate, it can also increase the efficiency of the P2P one-view MPVC by conserving the uplink capacity. The reason is that the clustered P2P

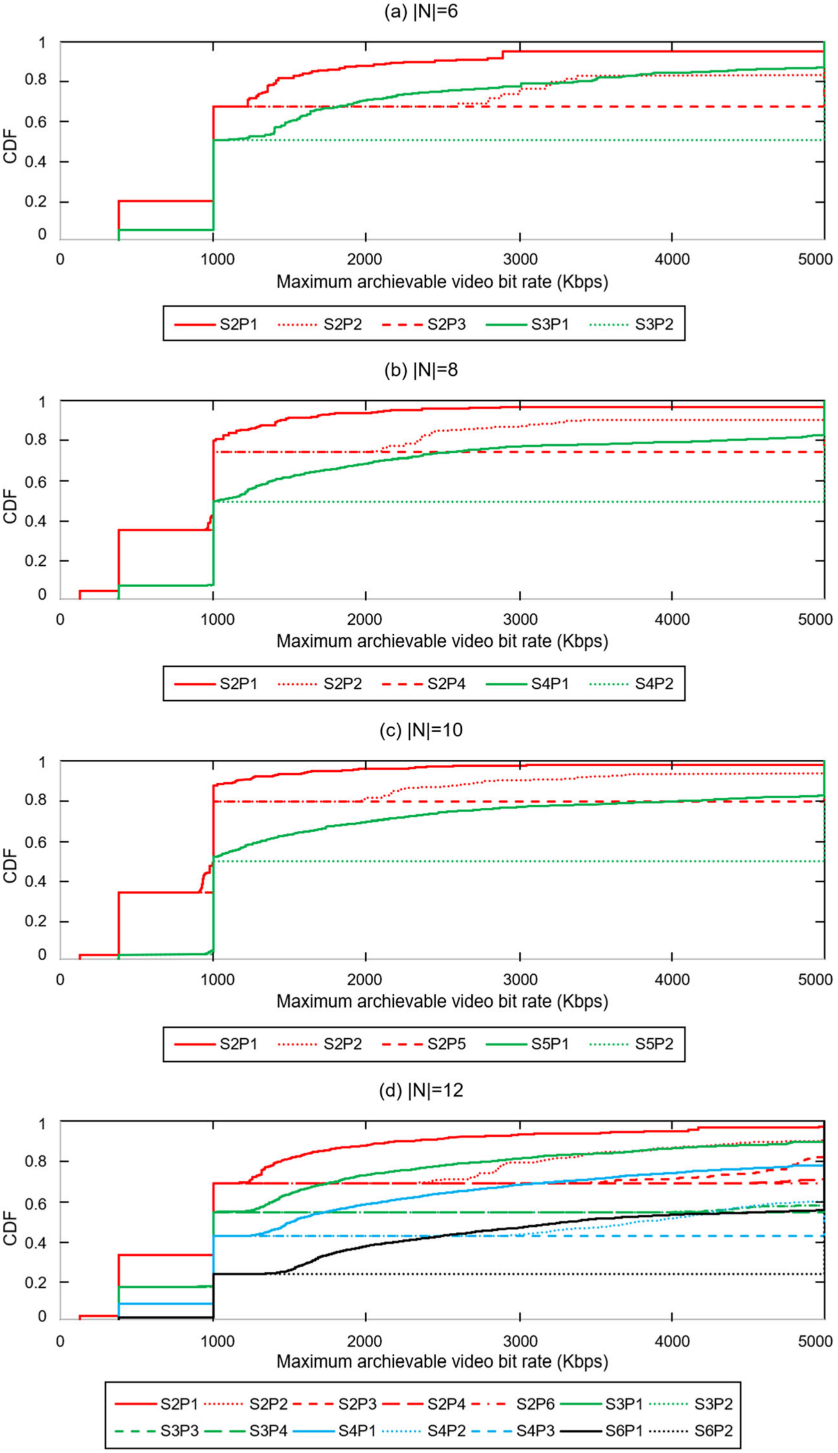


Fig. 4 Maximum achievable video bit rate. SnPm indicates that $|S| = \text{n}$ and $|P_V| = \text{m}$. Clustering is not applied in SnP1 cases

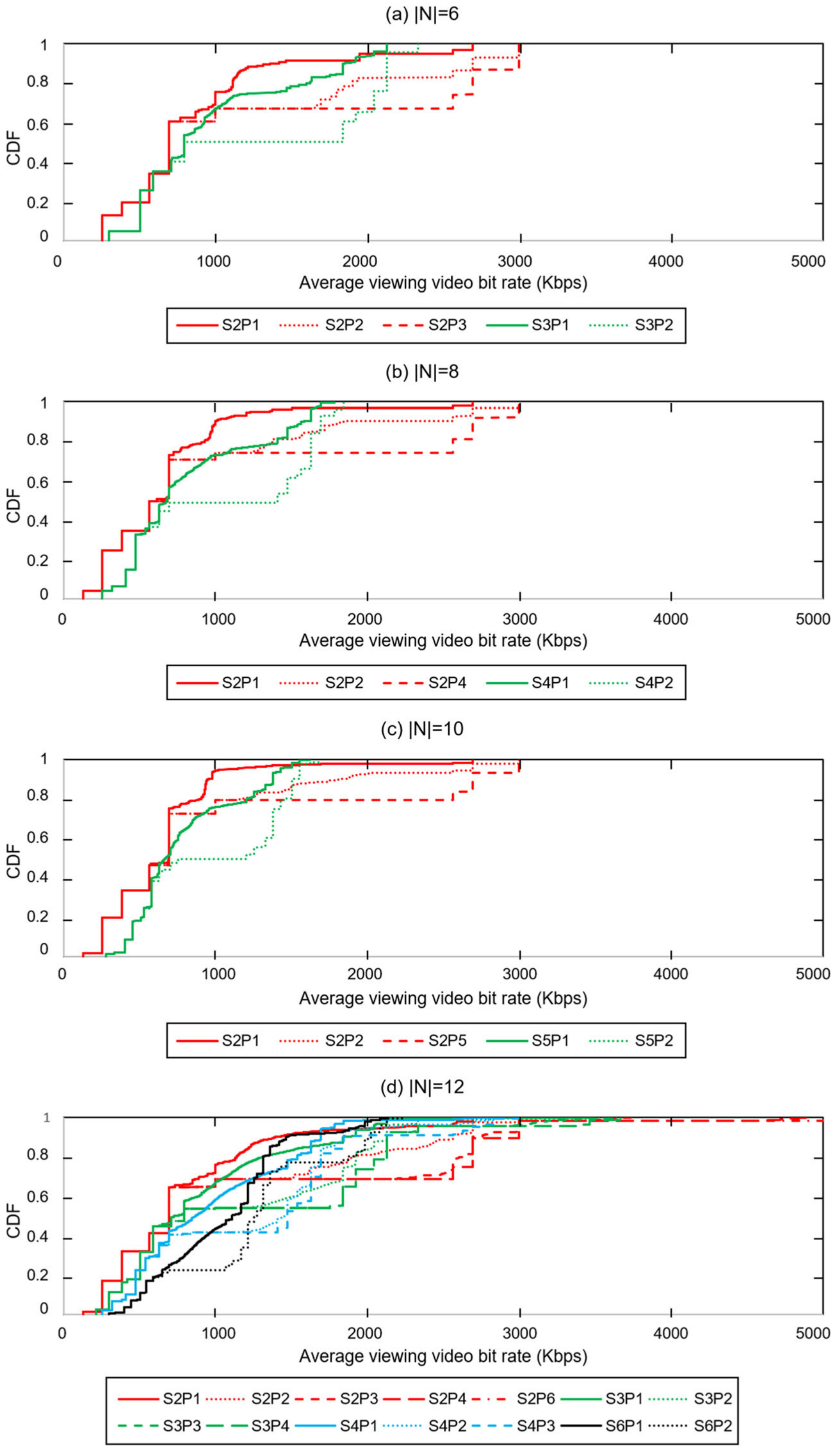


Fig. 5 Average video bit rate perceived by peers. SnPm indicates that $|S| = \mathrm{n}$ and $|P_V| = \mathrm{m}$. Clustering is not applied in SnP1 cases

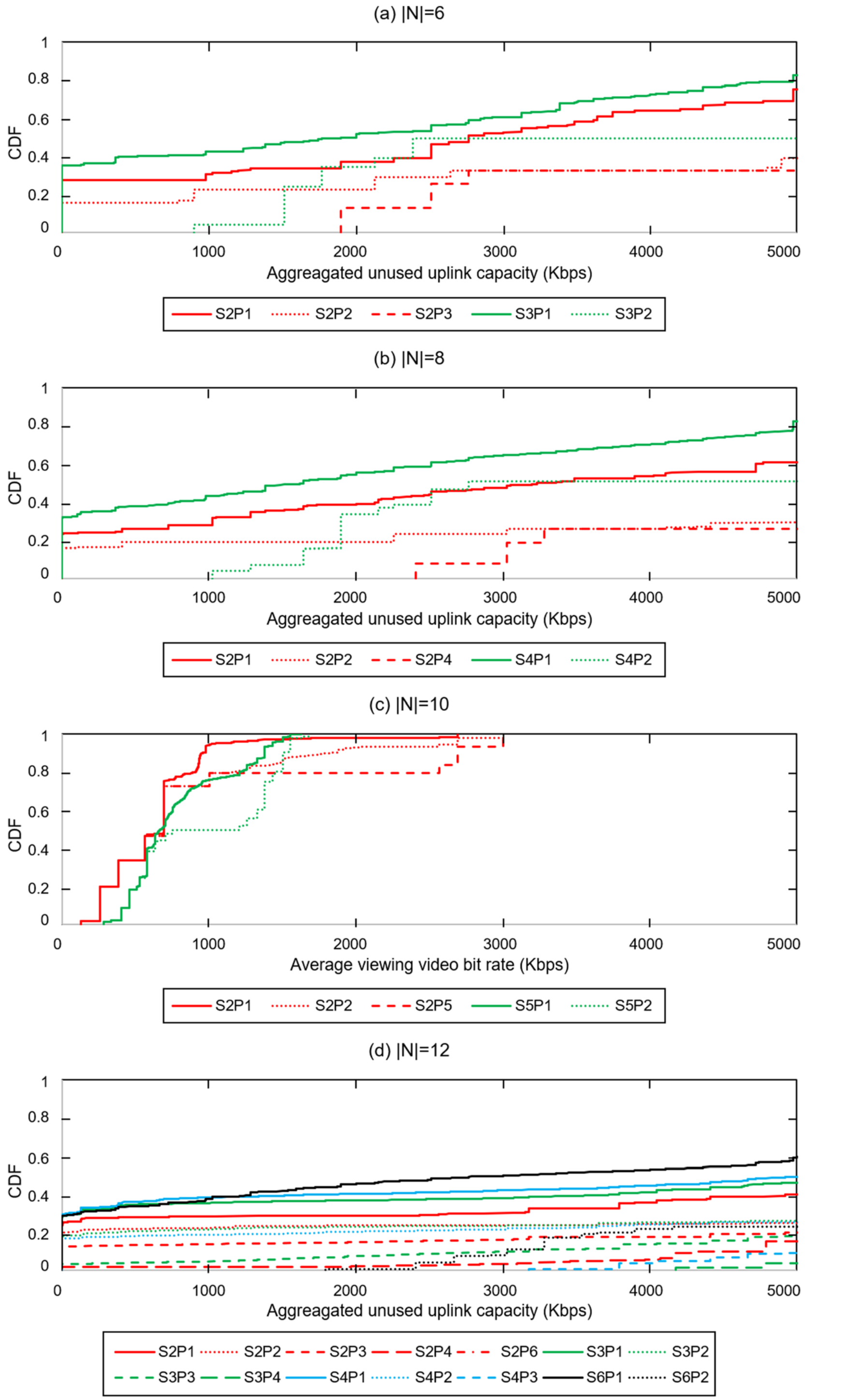


Fig. 6 Aggregated unused uplink capacity. SnPm indicates that $|S| = \mathrm{n}$ and $|P_V| = \mathrm{m}$. Clustering is not applied in SnP1 cases

one-view MPVC can offer a high video bit rate by consuming less amount of uplink capacity. $|S|$ affecting the aggregated unused uplink capacity is also observed. With an increase of $|S|$, sources can increase the bit rate of their video because they only share the video with few viewers. Then, the sources can allocate a large amount of uplink capacity to provide a high-quality video. The aggregated unused uplink capacity can be decreased because the number of peers with high burden is increased. However, even in such scenarios, clustering peers reduces the consumed uplink capacity. On the basis of the results from Figs. 4, 5 and 6, it is obvious that clustering peers results in scalable and efficient P2P one-view MPVC wherein every virtual peer has equal number of peers.

Apart from the P2P one-view MPVC with equal $|P_V|$, we performed an experiment with random $|P_V|$. This is the second case considered in Theorem 2. Specifically, every virtual peer possibly possesses a different number of peers unless the conferencing system is configured to have the same number of peers in every virtual peer. The configuration with random $|P_V|$ can also reflect the environment where multicast is not allowed even within a LAN segment; a virtual peer with $|P_V| = 1$ can be considered as such environment, since every peer will form an independent virtual peer if local multicast is not allowed. Unlike the previous experiments, the experiments with random $|P_V|$ had no specific combination between $|S|$ and $|P_V|$; hence, $|S|$ varied from 2 to $|N| - 1$. The configuration relies on the fact that every peer will join one subgroup and that clustering cannot be facilitated when every peer functions as a source. As observed in Figs. 4-6, three different aspects of the results were measured. To observe the improvement through peer clustering, the results were compared with the non-clustering case, which is a state-of-the-art system [28]. Figures 7, 8 and 9 depict the difference of the state-of-the art system [28] for each measurement aspect.

Figures 7 and 8 show a large difference between peer clustering and non-clustering with regard to the maximum achievable bit rate and the average bit rate perceived by all peers. As mentioned previously, a large number of sources result in high bit rate. Furthermore, the video bit rate can be maximized when all peers function as a source; each source can allocate all of its uplink capacity to offer a video to one viewer. This can be attributed to the fact that the difference between peer clustering and non-clustering is reduced with an increase in the number of sources. In some cases, no difference was observed between clustering with random $|P_V|$ and non-clustering.

However, in Fig. 9, less uplink capacity used to achieve the same video bit rate in these cases is evident. Thus, location-proximity-based clustering can be effective in achieving efficient P2P one-view MPVC.

Figure 9 depicts the CDF on the differences in the sum of unused uplink capacity. Evidently, the system conserving a large amount of uplink capacity is efficient, and clustering with random $|P_V|$ conserves a larger amount of uplink capacity than a system without clustering. Moreover, large amount of uplink capacity can be conserved as the number of peers in the conferencing system increased. The observation is important because it confirms that the P2P one-view MPVC can accommodate a large number of peers by forming virtual peers. The experiments with random $|P_V|$ also confirm that location-proximity-based clustering is effective in achieving scalable and efficient P2P one-view MPVC.

Apart from the effect on the video bit rate improvement and uplink capacity conservation, Section 5.1.1 discussed the approach of reducing the transmission delay, which is an important factor affecting the interactivity of the P2P one-view MPVC using the proposed clustering method. For each subgroup, specific viewers and helpers are selected based on the required

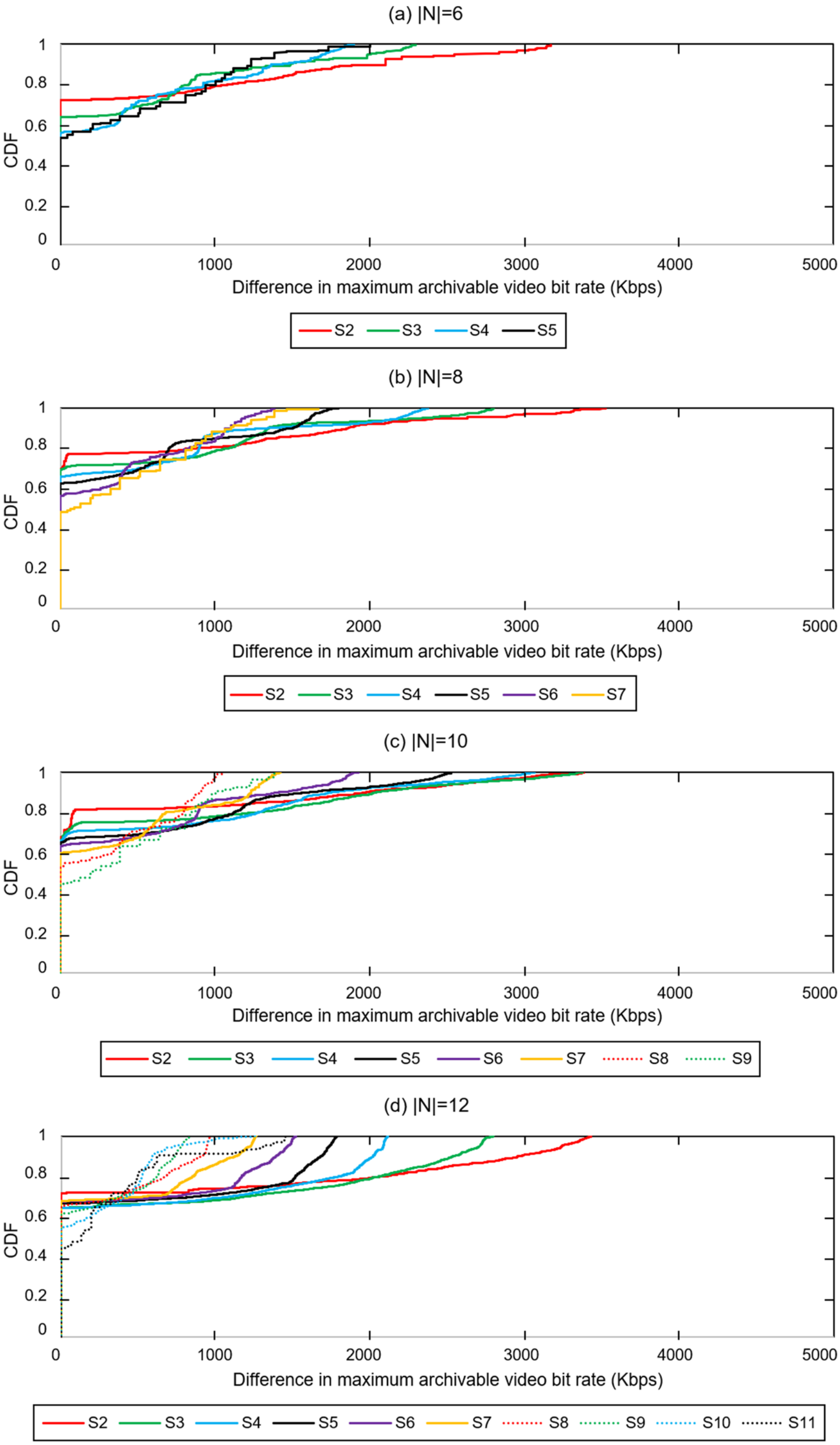


Fig. 7 Differences in the maximum achievable video bit rate. Sn indicates that $|S| = \mathrm{n}$

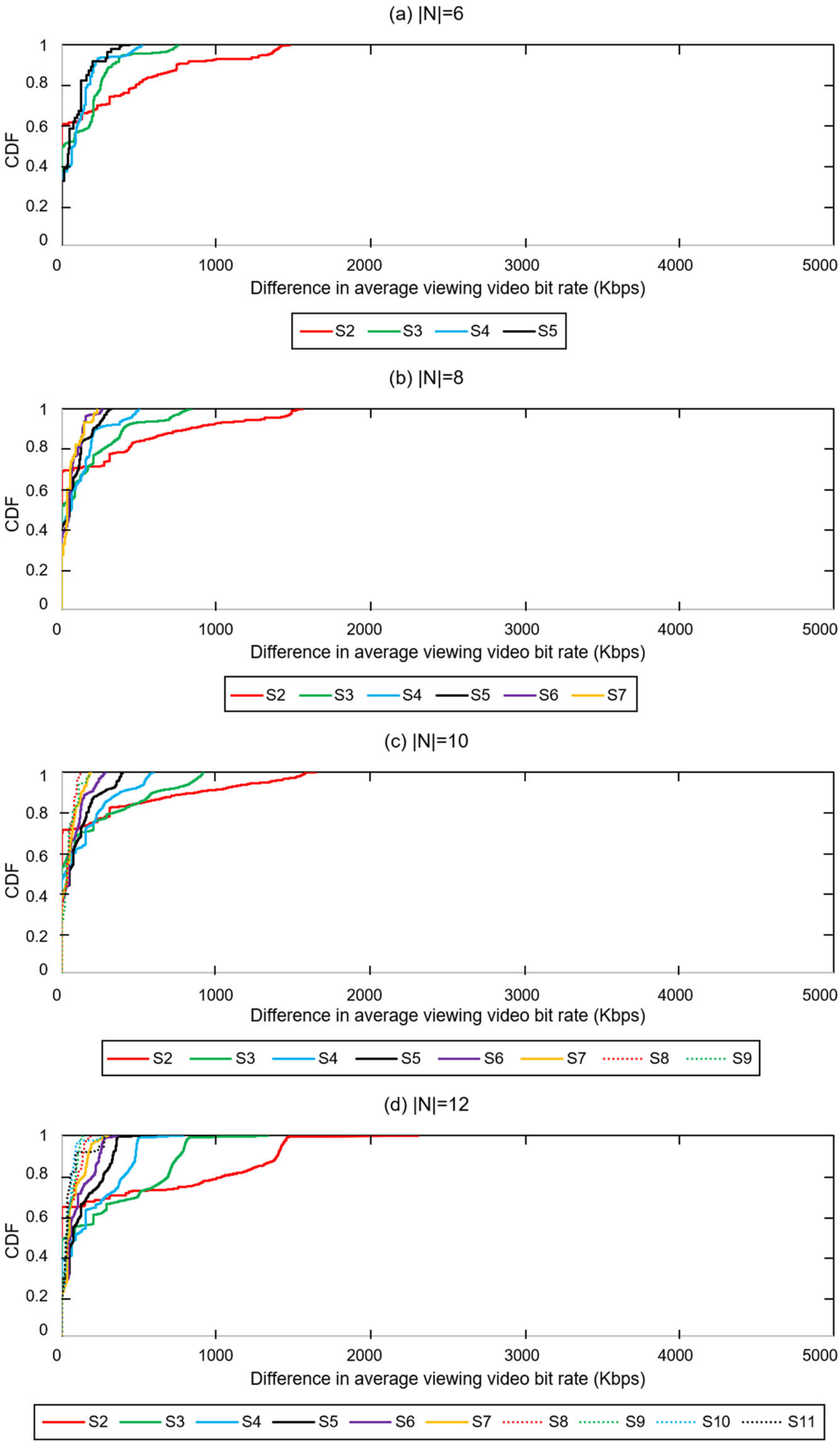


Fig. 8 Differences in the average video bit rate perceived by peers. Sn indicates that $|S| = \mathrm{n}$

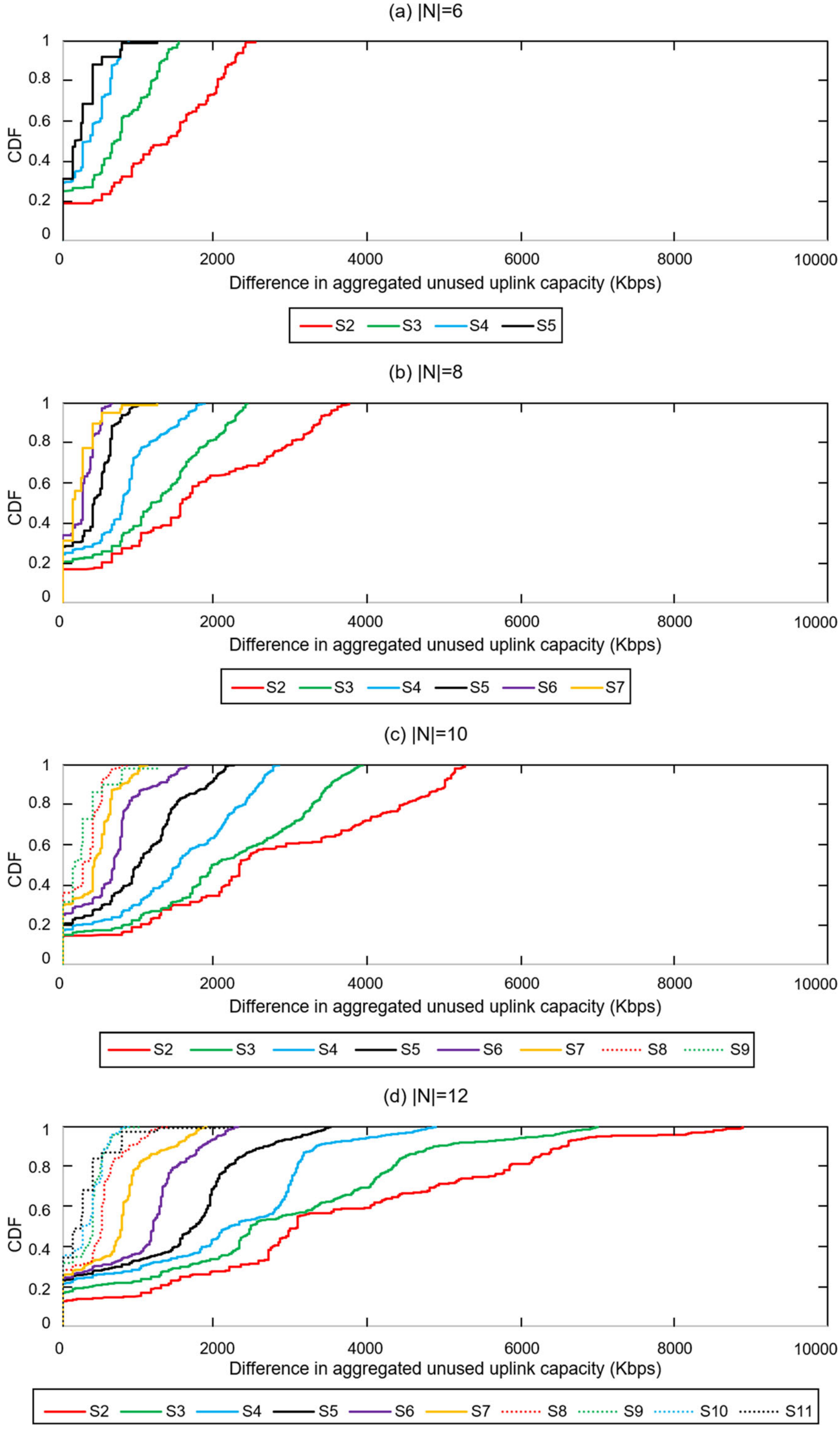


Fig. 9 Differences in the aggregated unused uplink capacity. Sn indicates that $|S| = n$

uplink capacity u_R^v and u_R^h, respectively. In the experiments, the target delay upper bound, d_T, was set to 1 s, and the fragment size, S_F, was set to 64 kb (=8 kB). Note that the target delay upper bound and fragment size were selected because the lowest uplink capacity of a peer was 128 Kbps [4]. Thus, the values can be configured differently with regard to the configuration of the peers' uplink capacity.

Figure 10 depicts the CDF of the maximum delay perceived by viewers. In every scenario, large $|P_V|$ resulted in a low perceived delay; hence, the transmission delay decreased with high $|P_V|$. In extreme cases wherein the number of virtual peers was equal to the number of sources, the observed largest delay is 0.5 s. This result is reasonable because the number of hops used for relaying substreams was only one in the extreme case, that is, transmission from a source to one virtual peer. Even the random $|P_V|$ cases show better performance than the observed performance in the non-clustered cases in every scenario. The results confirm that the proposed clustering method is effective in reducing the transmission delay, and thus, it can improve the interactivity among viewers.

This result is obtained because the selected viewers and helpers relay the received substreams to few viewers only as $|P_V|$ increases. Owing to the reduced responsibility of the viewers and helpers, they can allocate additional uplink capacity for relay. Therefore, the allocation of higher uplink capacity reduces the transmission delay.

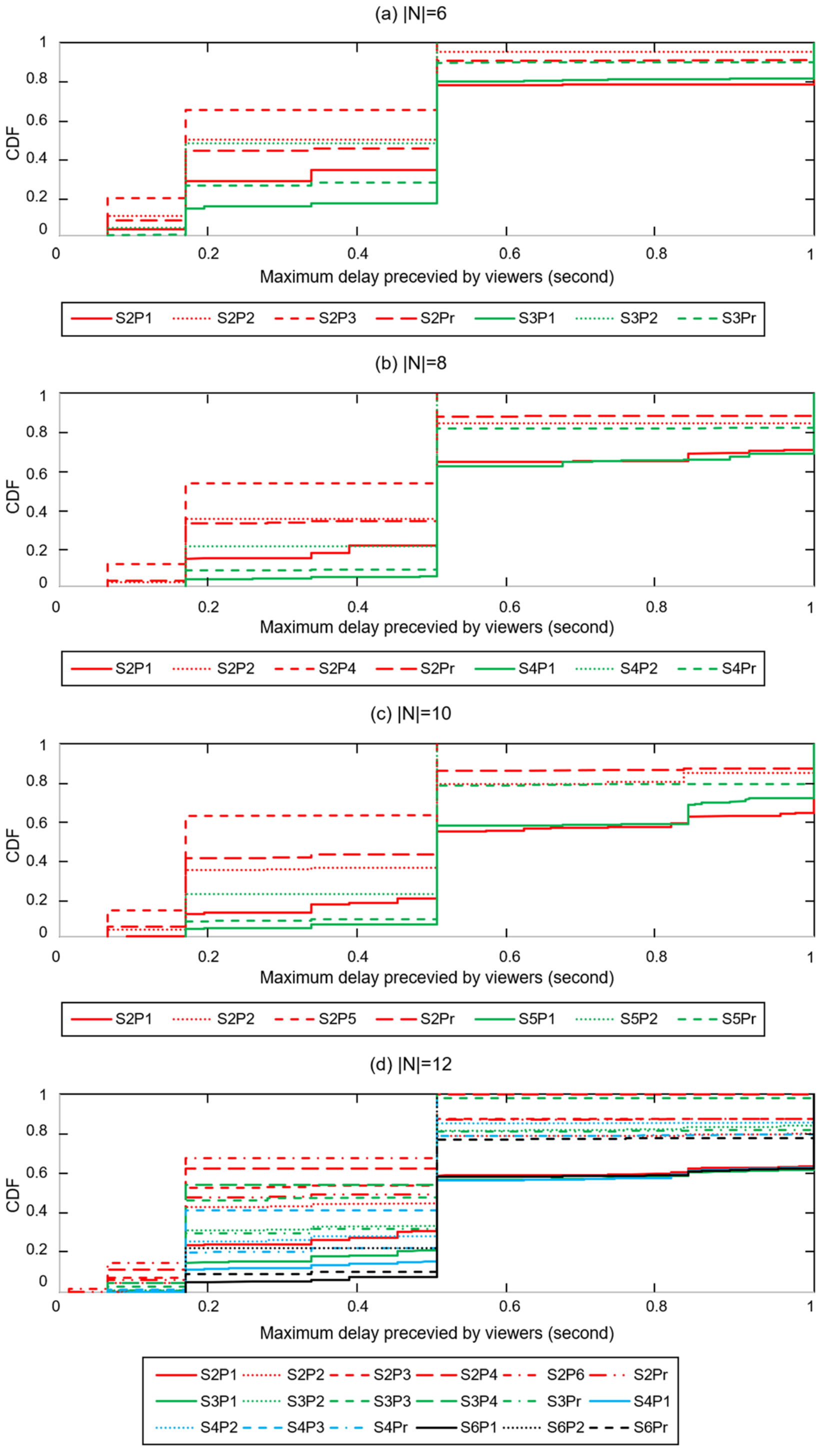
Apart from the transmission delay, the video bit rate and remaining uplink capacity were also evident. Section 5.1.1 describes how enforcing the uplink capacity requirement may produce different results regarding the improvement in the video bit rate and conservation of the uplink capacity because it excludes the peers that won't satisfy the enforced requirement.

Figures 11 and 12 plot the CDFs of the differences in the maximum achievable video and the average video bit rates for different peer and source scenarios, respectively. As shown in the figures, applying the uplink capacity requirement will result in similar performance when $|N|$ is set to 6.

However, the difference becomes large in the three other cases, namely, $|N|$=8, $|N|$=10, and $|N|$=12. This difference implies that applying the uplink capacity requirement achieves a low video bit rate in many experiments, as discussed in Section 5.1.1. However, the difference becomes minimal as $|P_V|$ increased. Considering that the number of virtual peers is highly related to the number of peers who belong to a virtual peer, $|P_V|$, many peers can be selected as viewers or helpers relaying substreams. Consequently, the achievable video bit rate can be increased. In extreme cases where the number of sources is equal to the number of virtual peers, the performance is the same regardless when the uplink capacity requirement is applied. The reason is that the number of hops for relaying substreams is equal to 1, and thus, the bit rate depends only on the uplink capacity of the sources. Furthermore, the random $|P_V|$ cases show better performance than the non-clustered ones in every scenario, suggesting that the proposed method is effective in actual environments.

Figure 13 plots the CDF of the difference in the aggregated unused uplink capacity of all peers. The figure shows that large uplink capacity can be maintained when the P2P MPVC satisfies the uplink capacity requirement, and increasing $|P_V|$ reduces the difference. The observation is reasonable because increasing $|P_V|$ results in video bit rate improvement that requires higher uplink capacity.

On the basis of this observation, the clustering method refined to enhance interactivity can also effectively improve the video bit rate and uplink capacity conservation in the P2P one-view MPVC.


(a) |N|=6

(b) |N|=8

(c) |N|=10

(d) |N|=12

Fig. 10 Maximum delay perceived by viewers. SnPm indicates that $|S| = \text{n}$ and $|P_V| = \text{m}$, where Pr indicates each virtual peer has random number of peers. Clustering is not applied when $|P_V| = 1$

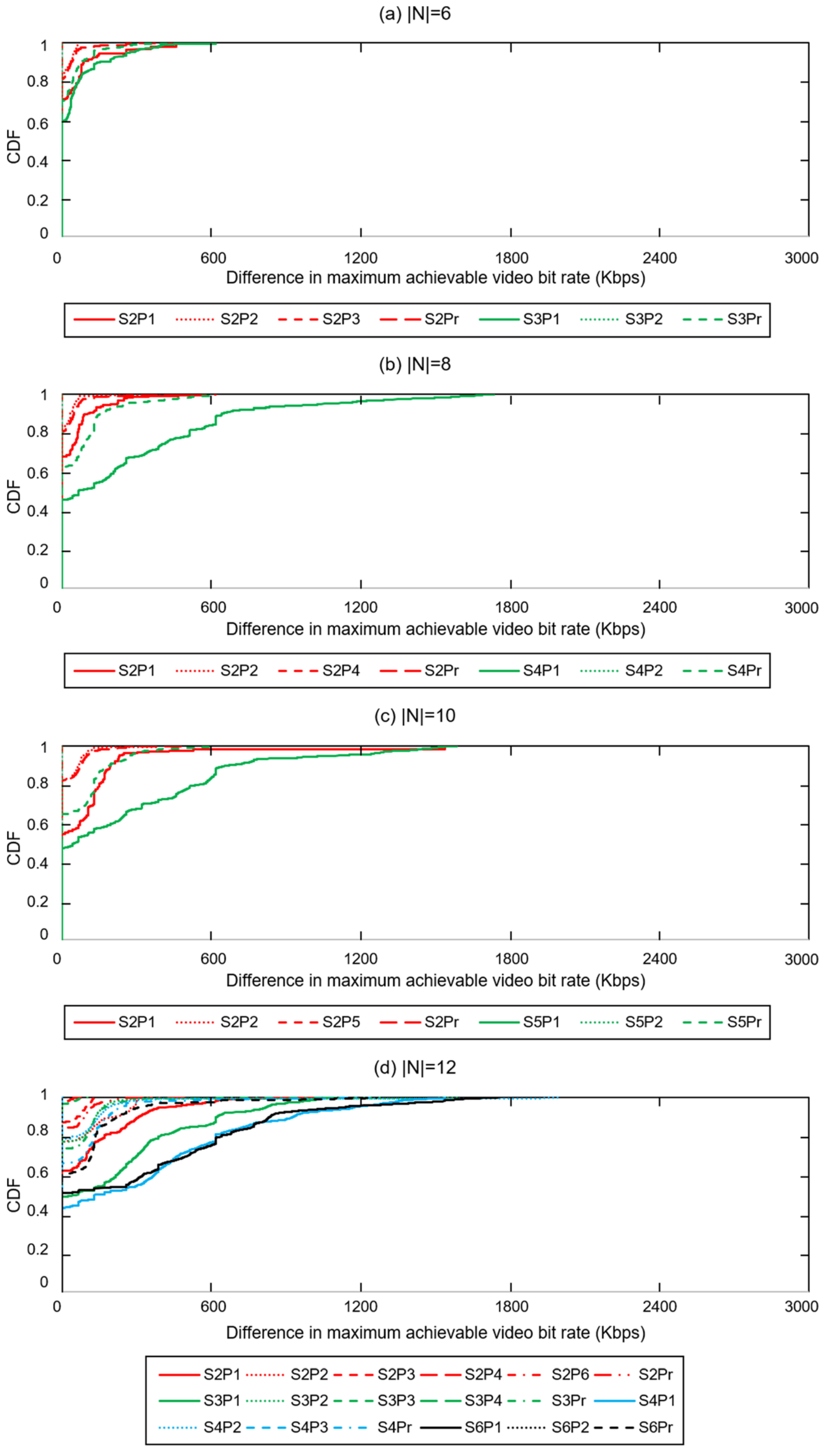


Fig. 11 Differences in the maximum achievable video bit rate. SnPm indicates that $|S| = \mathrm{n}$ and $|P_V| = \mathrm{m}$, where Pr indicates each virtual peer has random number of peers. Clustering is not applied when $|P_V| = 1$

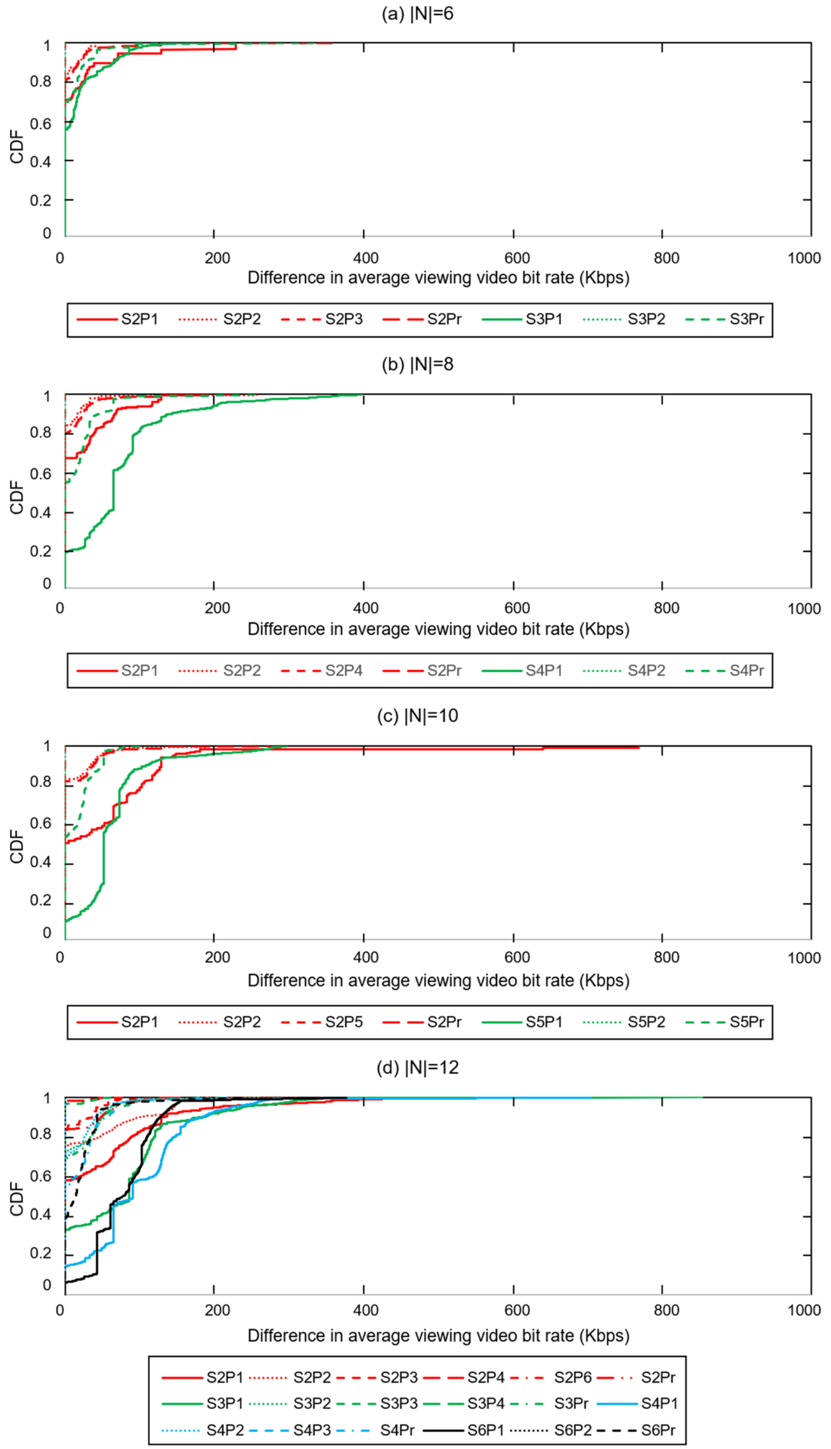


Fig. 12 Differences in the average video bit rate perceived by peers. SnPm indicates that $|S| = \text{n}$ and $|P_V| = \text{m}$, where Pr indicates each virtual peer has random number of peers. Clustering is not applied when $|P_V| = 1$

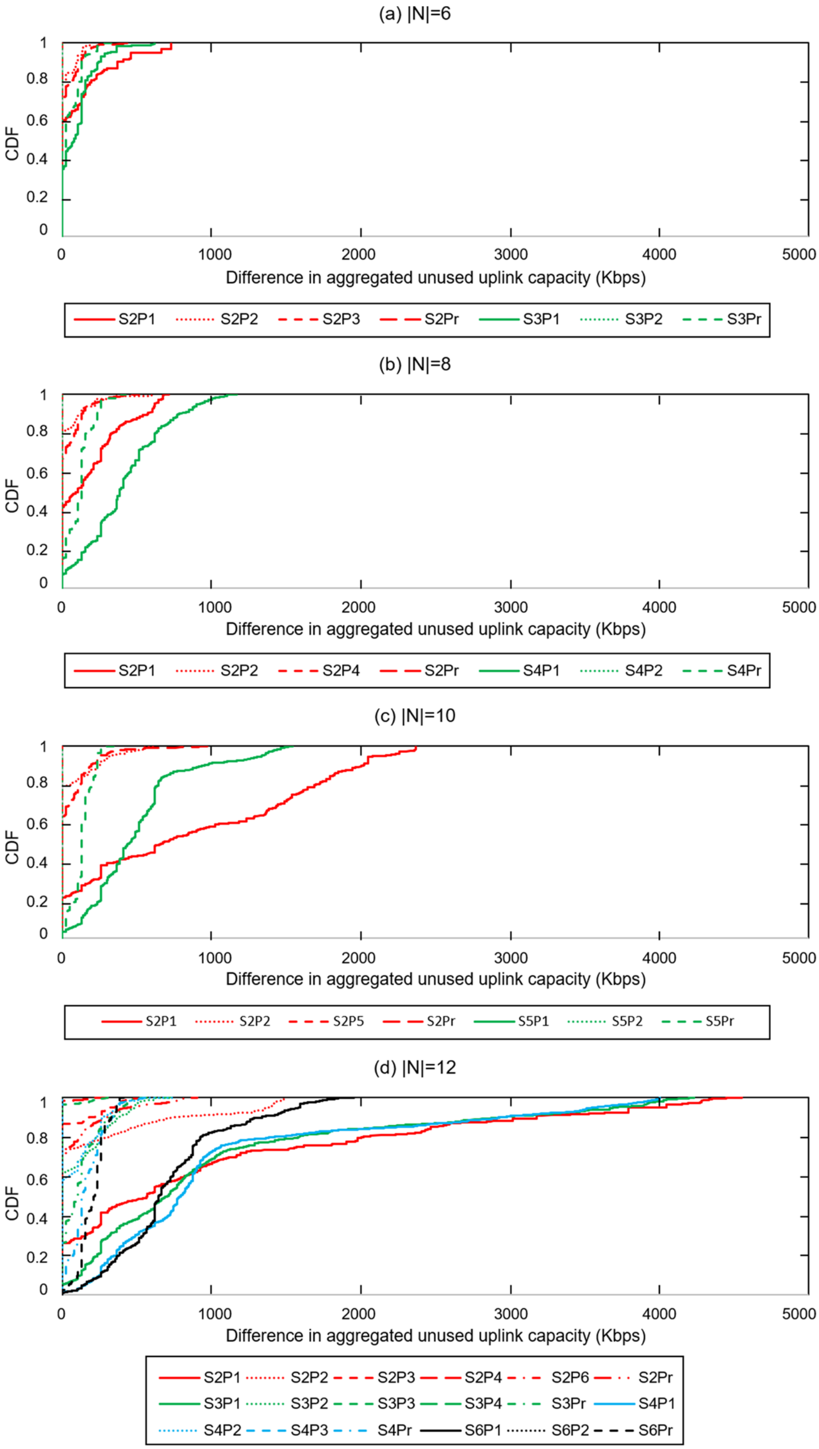


Fig. 13 Differences in the aggregated unused uplink capacity. SnPm indicates that $|S| = n$ and $|P_V| = m$, where Pr indicates each virtual peer has random number of peers. Clustering is not applied when $|P_V| = 1$

6.2 Effectiveness on multi-view video streaming

We also implemented C++ code for numerical evaluation to demonstrate the effectiveness of the proposed method on multi-view video streaming, and 4000 independent runs per experiment were executed. The total number of peers, |*N*|, is configured from 5000 to 25,000 with a step size of 5000. For each |*N*|, we varied the number of sources, |*S*|, from 2 to 4, and varied the number of peers in a virtual peer, $|P_V|$, from 1 to 5. Similar to the configuration of the prior experiments about the effectiveness on P2P one-view multiparty video conferencing, every virtual peer is assumed to have the same number of peers to observe the impact of $|P_V|$ on the performance of the system. $|P_V|$ is also configured to be random number to observe the validness on real environment. For heterogeneity, each peer has uplink capacity randomly selected among 128Kbps, 384Kbps, 1Mbps, and 5Mbps, according to the measurement study in [4]. Viewers are randomly allocated to each source in each scenario. We assumed that the multimedia stream is encoded as shown in Table 2 [3, 11].

To observe the uplink capacity conserved by the proposed method, we calculated the degree of conservation by averaging the ratios of the remaining uplink capacity to the maximum uplink capacity of every source, as follows:

$$\frac{\sum_{i \in S}\left\{u_i^s - (B_{S1} + B_{S2} + B_{S3})\right\}/u_i^s}{|S|} \tag{17}$$

The sum of ratio is divided by |*S*|, because each source has random number of viewers. In addition, every source is assumed to have an uplink capacity large enough to support $|G_i|$ viewers, which we obtained using

$$u_i^s = |G_i| \times b_i \tag{18}$$

, where b_i is the bit rate of the multimedia stream provided by source i. u_i^s means the required uplink capacity when source i sends multimedia streams to every viewer directly. Thus, the conservation degree represents how much the system is efficient and scalable compared to client-server system and non-clustered case. Higher conservation degree indicates that sources conserved more uplink capacity. In multi-view video streaming service, certain sources may not have any viewer. Such sources do not have an uplink capacity according to (18), and we thus excluded such sources in (17). Figure 14 illustrates the conservation degree for each scenario.

It is observed that the conservation degrees of clustered cases are around 0.99, while non-cluster cases show the conservation degrees under 0.8. We can also observe that the conservation degree is slightly improved with the number of sources. The reason is that sources may have a smaller number of viewers as the number of sources is increased. It is remarkable that the proposed method results in huge improvement of the conservation degree under every scenario. In some cases, every source spends just 2 Mbps to accommodate 25,000 viewers. This indicates that the contributed uplink capacity from the peers, including viewer and

Table 2 Bit rate configuration

Quality	Bit rate
SD	2 M (CBR, H.264)
HD	8 M (CBR, H.264)
UHD	18 M (CBR, H.265)

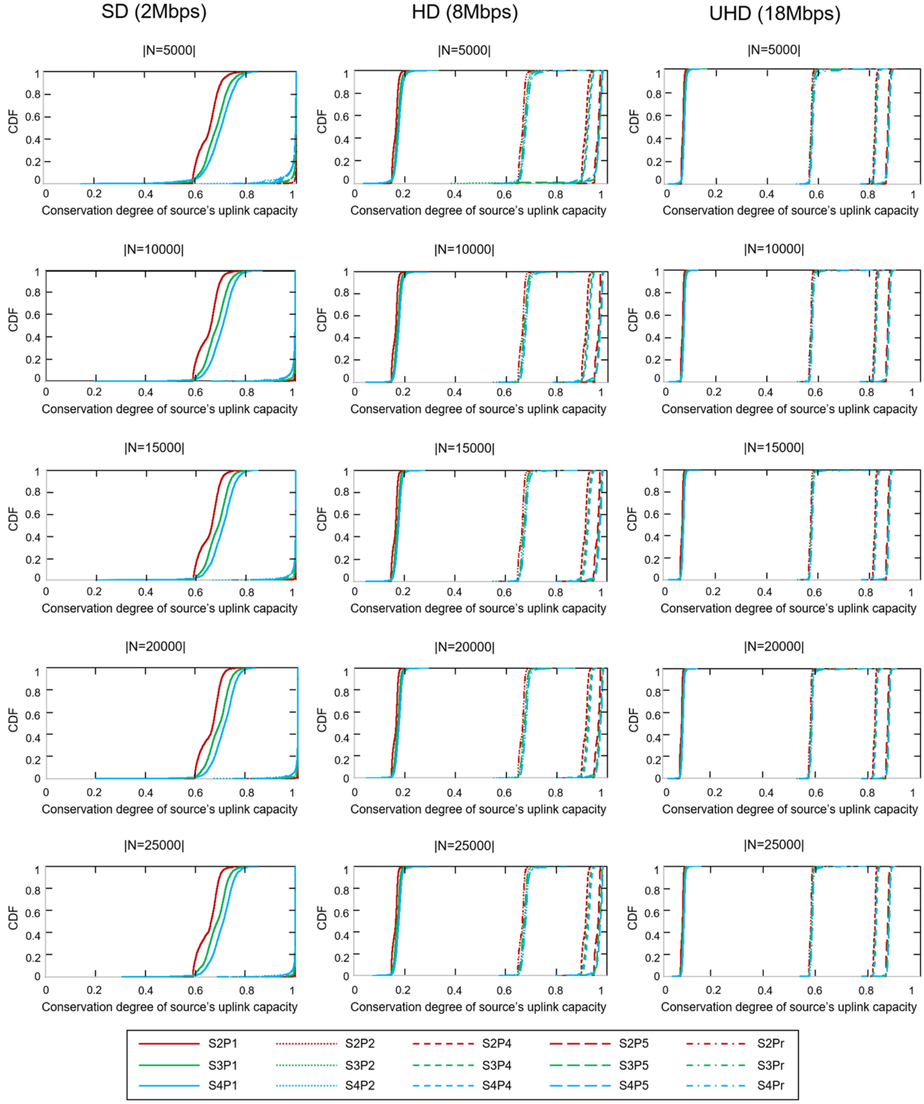


Fig. 14 Conservation degree of source's uplink capacity. SnPm indicates that $|S| = \mathrm{n}$ and $|P_V| = \mathrm{m}$, where Pr indicates each virtual peer has random number of peers. Clustering is not applied when $|P_V| = 1$

helpers, is sufficient to distribute SD multimedia streams. However, different results are observed when the sources provide HD and UHD multimedia streams. The uplink capacity required to distribute HD and UHD multimedia streams exceeds the contribution from peers. Thus, sources need to spend more uplink capacity to deliver substreams along the third delivery route, which sources send substream to viewers directly. It is observed that sources need to spend almost the entire uplink capacity to provide streams when $|P_V| = 1$; conservation degree is under 0.1. However, the proposed method remarkably enhances the degree of conservation, as shown in HD and UHD cases of Fig. 14. It is also remarkable that random $|P_V|$ cases show the conservation degree similar to that of $|P_V| = 2$ cases. The results imply that the proposed method can be highly effective in real world.

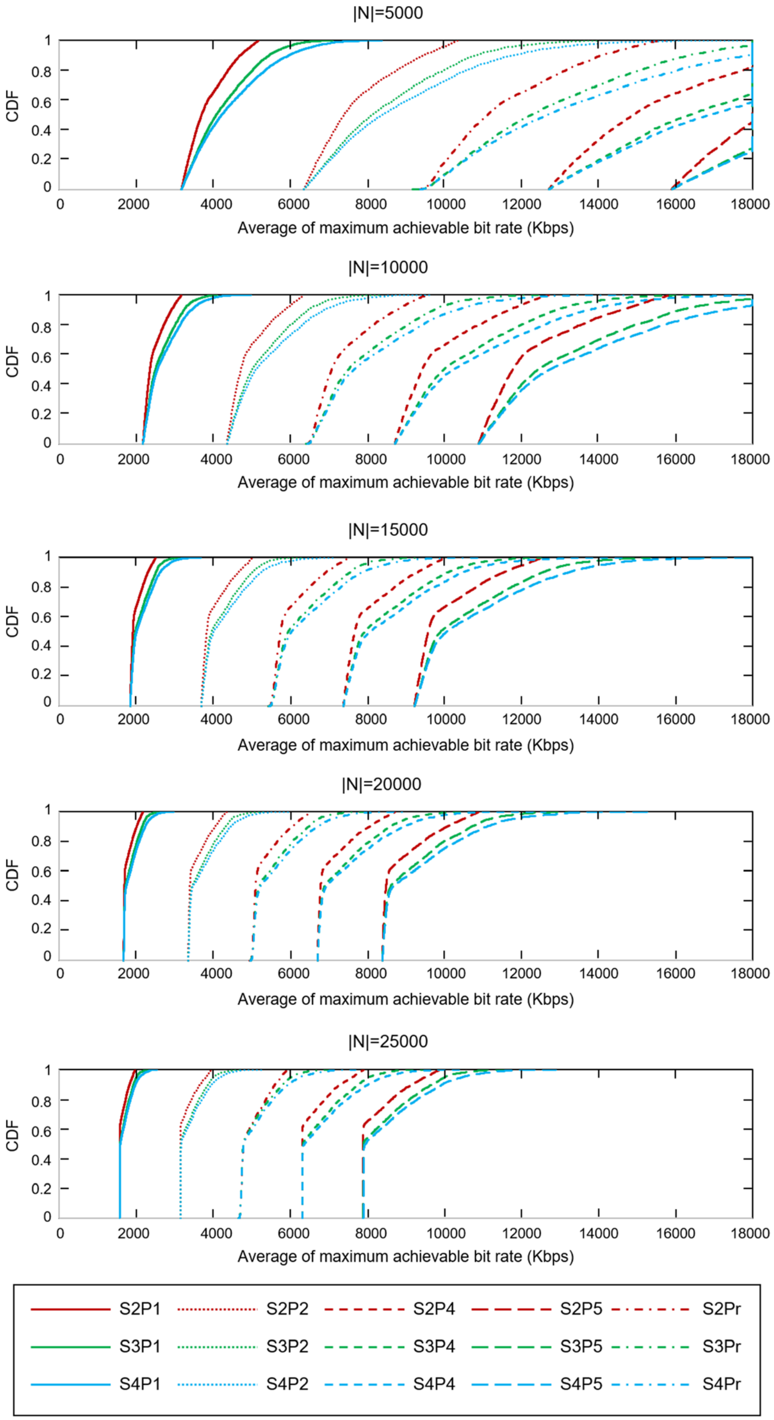


Fig. 15 Average of maximum achievable bit rate under limited uplink capacity of sources. SnPm indicates that $|S| = \mathrm{n}$ and $|P_V| = \mathrm{m}$, where Pr indicates each virtual peer has random number of peers. Clustering is not applied when $|P_V| = 1$

In the above experiments, we assumed that every source has an abundant uplink capacity, as indicated in (18). In real environment, each source cannot have such abundant uplink capacity which is linear in the number of viewers. For practical results, we limited the upload capacity of the sources to 10 Gbps. We then conducted the experiments with 4000 independent runs per each scenario for observing the maximum achievable bit rate under the condition. Then we calculated the average of 4000 results for each scenario. Figure 15 shows the average of maximum achievable bit rate for each scenario.

The results show that the maximum achievable bit rate of the multimedia streams is improved as $|S|$ is increased. With more sources, it is likely that viewers consume the multimedia streams from different sources, and thus each source may have fewer viewers. As a result, the burden on sources and their viewers can be alleviated. With the alleviated burden, the maximum achievable bit rate can be increased. Figure 15 also shows the remarkable effectiveness of the clustering method. For example, in the scenario which $|N|$=15,000 with two sources, the average of maximum achievable bit rate of non-clustered cases is under 2600 Kbps, while all other cases, which are $|P_V|$=2, $|P_V|$=4, $|P_V|$=5, and random $|P_V|$, can achieve much higher bit rate; atmost 12Mbps when $|P_V|$=5. It is remarkable that random $|P_V|$ cases show much higher bit rate than non-clustered cases in every scenario and also can achieve higher bit rate than that of $|P_V|$=2 cases. From the results, we can expect that the clustering method can be highly effective in real environment. We can also observe that the average maximum achievable bit rate is decreased with larger $|N|$. The reason is the fact that the uplink capacity of peers affects the bit rate of substreams. Let assume that $|G_i|$ viewers with the same uplink capacity, C, are grouped into $|VP_i|$ virtual peers, and source i has infinite uplink capacity; infinite uplink capacity is assumed to focus on the uplink capacity of viewers. The bit rate of substream is $C/(|VP_i|-1)$, according to Algorithm 3, and the achievable bit rate of the multimedia stream is $|G_i| \times C/(|VP_i|-1)$. In fixed $|P_V|$ cases, the achievable bit rate of multimedia streams is $|G_i| \times C/\left(\frac{|G_i|}{} |P_V|-1\right)$. Thus, the achievable bit rate will be decreased as $|G_i|$ is increased. It is easily inducible that higher $|P_V|$ will alleviate the tendency of decreasing the achievable bit rate, as depicted in Fig. 15. $|P_V|$ of each virtual peer is at least 1, in random $|P_V|$ cases. Thus, random $|P_V|$ cases also follow the similar tendency of decrease. The effectiveness of the clustering method is obvious. With a certain number of viewers, the clustering method can be effective in achieving higher bit rate, which directly affects the quality of multimedia streams. With a certain bit rate of multimedia streams, the clustering method can provide higher scalability by accommodating more viewers.

7 Conclusion and future work

In this study, we presented location-proximity-based clustering for P2P multimedia streaming with multiple sources to achieve scalability. Peers in a LAN who participate in the same P2P network can be grouped into a logical entity, referred to as virtual peer, to reduce the peer's responsibility. We described three methods herein: for identifying the location of peers, for forming virtual peers, and for balancing the relay burden among peers in a virtual peer. This work further provided an analysis on the maximum achievable bit rate to confirm the improvement obtained via location-proximity-based clustering. We introduced two applications of multimedia streaming with multiple sources, which are P2P one-view multiparty video conferencing and multi-view video streaming, and discussed considerations for applying the proposed method to the applications. The experiment results confirmed that location-proximity-based clustering is effective for both applications in improving the maximum achievable video bit rate and average viewing video bit rate and in conserving the uplink capacity of the peers. In addition, the proposed clustering method effectively reduces the transmission delay for P2P one-view multiparty video conferencing, which directly affects the interactivity. Therefore, the location-proximity-based clustering method is effective in achieving a scalable and efficient P2P multimedia streaming with multiple sources.

The future work will extend this study with consideration on various types of service. Depending on the type of service, P2P network may not be stable because of user behavior changing video during service. Further study with consideration on peer dynamics will show the effectiveness of the clustering method on various services.

Code availability Not applicable.

Funding This work was supported by the ICT R&D programs of MSIP/IITP. [2016–0-00192, Standards development for service control and contents delivery for smart signage services].

[2019-0-00456, Development of standard technology on hybrid P2P networking protocol for IoT data streaming and enhancement of transfer performance of block chain service].

Data availability Not applicable.

Declarations

Conflicts of interest/competing interests Author have no conflict of interest to declare. All authors have seen and approved the submission of manuscript.

References

1. AkkuŞ İE, Özkasap Ö, Civanlar MR (2011) Peer-to-peer multipoint video conferencing with layered video. J Netw Comput Appl 34:137–150
2. An Z, Liu HY, Zhang H (2011) Audio-video conference systems research based on P2P and Multicast Proc. International Conference on Computational and Information Sciences: 809–812
3. Bae SH, Kim J, Kim M, Cho S, Choi JS (2013) Assessments of subjective video quality on HEVC-encoded 4K-UHD video for beyond-HDTV broadcasting services. IEEE Trans Broadcast 59:209–221
4. Bharambe AR, Herley C, Padmanabhan VN (2006) Analyzing and improving a BitTorrent networks performance mechanisms. Proc. IEEE INFOCOM
5. BitTorrent (2020) https://www.bittorrent.com
6. Chen X, Yang C (2009) Banacast: a fast adaptive application layer multicast network for P2P Live Streaming. Proc. International Conference on Information Science and Engineering: 149–153.
7. Chen M, Ponec M, Sengupta S, Li J, Chou P (2012) Utility maximization in peer-to-peer systems with applications to video conferencing. IEEE/ACM Trans Networking 20:1681–1694
8. Google+ (2019), https://plus.google.com
9. IEEE (2017) HTTP. Live Streaming IEEE RFC 8216
10. ISO/IEC (2014) Dynamic adaptive streaming over HTTP (DASH) - part 1: media presentation description and segment formats. ISO/IEC 23009-1:2014
11. ITU-T (2008) G.1080, Multimedia quality of service and performance – generic and user-related aspects
12. ITU-T (2010) Information technology - Relayed multicast protocol: Specification for N-plex group applications. ITU-T Recommendation X.603.2
13. ITU-T (2015) Managed peer-to-peer (P2P) communications: Functional architecture. Recommendation ITU-T X.609
14. Kumar R, Liu Y, Ross K (2007) Stochastic fluid theory for P2P streaming systems. Proc. IEEE INFOCOM

15. Kurdoglu E, Liu Y, Wang Y (2016) Dealing with user heterogeneity in P2P multi-party video conferencing: layered distribution versus partitioned simulcast. IEEE trans Multimedia 18:90–101
16. Li J, Chou PA, Zhang C (2004) Mutualcast: An efficient mechanism for content distribution in a P2P network. Proc, SIGCOMM Asia Workshop
17. Liu Y, Guo L, Li F, Chen S (2009) A case study of traffic locality in internet P2P live streaming systems. Proc. IEEE 29th International Conference on Distributed Computing Systems: 423–432.
18. Manzillo MP, Ciminiera L, Marchetto G, Risso F (2012) CLOSER: a collaborative locality-aware overlay SERvice. IEEE Trans Parallel Distrib Syst 23:1030–1037
19. National Institute of Standards and Technology (1995) Secure hash standard (SHS). Federal Information Processing Standards Publication FIPS PUB:180–181
20. Ponec M, Sengupta S, Chen M, Li J, Chou PA (2011) Optimizing multi-rate peer-to-peer video conferencing applications. IEEE Trans Multimedia 13:856–868
21. Shen H, Liu G, Ward L (2015) A proximity-aware interest-clustered P2P file sharing system. IEEE Trans. Parallel Distrib Syst 26:1509–1523
22. Skype (2020), http://www.skype.com
23. Tu X, Jin H, Liao X (2008) Nearcast: a locality-aware P2P live streaming approach for distance education. ACM Transactions on Internet Technology (TOIT) 8:1–23
24. WebRTC (2020), https://webrtc.org
25. Wu W, Mao H, Wang Y, Wang J, Wang W, Tian C (2017) CoolConferencing: enabling robust peer-to-peer multi-party video conferencing. IEEE Access 5:25474–25486
26. Xu Y, Yu C, Li J, Liu Y (2014) Video telephony for end-consumers: measurement study of Google+, iChat, and Skype. IEEE/ACM Trans Networking 22:826–839
27. X. Zhang, J. Liu, B. Li and Y. S. P. Yum (2005) CoolStreaming/DONet: a data-driven overlay network for peer-to-peer live media streaming. Proc. IEEE INFOCOM
28. Zhao Y, Liu Y, Chen C, Zhang J (2014) Enabling P2P one-view multiparty video conferencing. IEEE Trans. Parallel Distrib. Syst. 25:73–82

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Affiliations

Changkyu Lee[1,2] **· Shin-Gak Kang**[1,2] **· Anand Nayyar**[3]

Shin-Gak Kang
sgkang@etri.re.kr

Anand Nayyar
anandnayyar@duytan.edu.vn

[1] Department of ICT, University of Science and Technology (UST), Daejeon, Republic of Korea

[2] Protocol Engineering Center (PEC), Electronics and Telecommunications Research Institute (ETRI), Daejeon, Republic of Korea

[3] Graduate School, Faculty of Information Technology, Duy Tan University, Da Nang 550000, Vietnam