

Enabling Peer-to-Peer One-view Multi-party Video Conferencing

Yongxiang Zhao^{†,‡}, Yong Liu[†], Changjia Chen[‡], Hongluan Liao^{*}, and Jianyin Zhang^{*}

[†]Electrical & Computer Engineering, Polytechnic Institute of NYU, Brooklyn, NY, USA

[‡]School of Electrical and Information, Beijing Jiaotong university, Beijing, CHINA

^{*}Research Institute of China Mobile, Beijing, CHINA

Abstract—Multi-Party Video Conferencing (MPVC) facilitates realtime group interaction between users. While P2P is a natural delivery solution for MPVC, a peer often does not have enough bandwidth to delivery her video to all other peers in the conference. Recently, we have witnessed the popularity of one-view MPVC, where each user only watches full video of another user. One-view MPVC opens up the design space for P2P delivery. In this paper, we explore the feasibility of a pure P2P solution for one-view MPVC. We characterize the video source rate region achievable through video relays between peers. For both homogeneous and heterogeneous MPVC systems, we establish tight universal video rate lower bounds that are independent of the number of peers, the number of video sources, and the specific viewing relations between peers. We further propose P2P video relay designs to approach the maximal video rate region. Through numerical simulations, we verified that the derived lower bounds are indeed tight bounds, and the proposed bandwidth allocation algorithm can achieve a close-to-optimal peer upload bandwidth utilization. Our results demonstrate that P2P is a promising solution for one-view MPVC. Insights obtained from our study can be used to guide the design of practical P2P one-view MPVC systems.

I. INTRODUCTION

The Internet has fundamentally changed the way people communicate, from emails, text-messages, blogs, tweets, to Voice-over-IP calls, etc. We are now experiencing the next big change: *Video Telephony*. Although video telephony was originally conceived in 1920s, largely due to its stringent bandwidth and delay requirement, it has little success, especially in the end-consumer market, until very recently. The proliferation of video-capable consumer electronic devices and the penetration of increasingly faster residential network accesses paved the way for the wide adoption of video telephony. More and more users are enjoying video calls offered by GTalk, Skype, MSN messenger, etc. Video telephony is more than just two-party video call. It also supports Multi-Party Video Conferencing (MPVC), which facilitates realtime group interaction between users. P2P is a natural delivery solution for MPVC where users transmit their video and voice directly among themselves. But the major challenge for P2P MPVC is that users alone may not have enough upload bandwidth to transmit all their videos.

Skype [18] offers MPVC service to its paid premium customers. Our recent measurement study shows that in a Skype MPVC, while voice is still transmitted using P2P, video of a user is first uploaded to a server, then relayed to all other users in the conference. This design choice is due to the fact

that in an *all-view* MPVC, where each user watches videos of all other users, the aggregate video upload workload increases quadratically with the number of users, while the aggregate upload capacity available on users only increases linearly. Pure P2P is obviously not a self-scalable solution. Hybrid peer-assisted solutions have been studied recently [13], [4]. Another concern for all-view MPVC is that, even though servers can provide abundant upload bandwidth, the downlink of a user might not be able to sustain high-quality video streams from all other users. More recently, Google+ [8] offers a free *one-view* MPVC service: each user can only choose one user to watch at high video quality, and receives all other users' videos at the minimum video quality.^{1 2} Our measurement study shows that Google+'s one-view MPVC is still implemented as a pure server-based solution: a user chooses a dedicate server as her MPVC proxy, uploads her voice and video data to the proxy, and downloads voice and video data of other users from the proxy. Such a server-centric solution not only incurs high server cost, but also introduces additional delays in video/voice relay.

In one-view MPVC, the aggregate video download workload is reduced to be proportional to the number of users. The aggregate peer upload bandwidth can now keep up with the aggregate video upload workload. It is therefore tempting to develop a pure P2P solution for one-view MPVC. Such a solution not only eliminates the server cost, but also can explore user locality better to achieve shorter delays, which is critical to facilitate realtime user interactions. In addition, P2P MPVC is an attractive solution to set up ad-hoc MPVC not subject to centralized management and monitoring.

P2P MPVC is admittedly more challenging than P2P VoIP and P2P video streaming. Compared with voice, video is much more bandwidth-demanding. Compared with video streaming, MPVC is much more sensitive to delay. While seconds of buffering delay is often tolerable in video streaming, in MPVC, user QoE degrades significantly if the delay goes over 500 milli-seconds. In addition, P2P relay design in MPVC is more complicated than in video streaming. In P2P video streaming, a set of peers watching the same video source form

¹A user can dynamically choose her full-quality video source based on her current interest. By default, the system will send full video of the user currently speaking to users not specifying their interests.

²In our recent measurement study of google+, we found that the size of a small video is only about 6% of a full video.

a swarm and relay video to each other. Due to the common video interest, video relay between peers are mostly driven by their bandwidth availability. In P2P MPVC, peers have diverse viewing interests. Each peer is a potential video source watched by other peers, and at the same time is watching another source. The viewing relations between peers are intrinsically *entangled*. More challengingly, peers' viewing interests are constantly driven by various conferencing dynamics, such as user voice and gesture activities, appearance of new objects, and topic switching, etc. Video relays between peers have to be adaptive to the entangled and dynamic viewing relations.

In this paper, we explore the feasibility of P2P one-view MPVC by characterizing its capacity region through analysis and numerical simulations. Motivated by Google+, we assume a peer only watches one full video of another peer at any given time, and can switch her view interest at will. We assume that voices and small videos of peers are delivered using some traditional P2P technique and only focus on the P2P delivery of full videos between peers.³ We further assume that peers in the same conference are cooperative and relay videos for each other. To maintain good delay performance, P2P video relay is limited to two-hops. The contributions of our study is four-fold:

- 1) We propose a P2P relay framework for one-view MPVC. We characterize the video rate capacity region for homogeneous and heterogeneous one-view MPVC. We study the optimal P2P relay design to maximize the aggregate video quality. We also propose rate allocation scheme to achieve the *max-min fairness* between video sources.
- 2) We establish several *universal video rate lower bounds* for P2P one-view MPVC that are *independent* of the viewing relations between peers. For homogeneous one-view MPVC with normalized peer upload bandwidth of 1, we show that each source is guaranteed to achieve the video rate of $5/6$, that is also independent of the size of MPVC and the number of sources.
- 3) For heterogeneous MPVC with the normalized average peer upload bandwidth of 1, we show that the guaranteed video rate for source i with upload bandwidth u_i is $\min(u_i, \gamma)$, where $\gamma = \max(\frac{2}{3}, \frac{|N|}{|N|+|S|})$ with $|N|$ being the number of peers and $|S|$ being the number of sources. The lower bound can be improved to $3/4$ if all sources' upload bandwidth is above the average. We further show that the derived lower bounds are tight for homogeneous and heterogeneous systems.
- 4) We develop peer bandwidth allocation algorithms that efficiently utilize peers' upload bandwidth to approach the maximal video rate region. Through numerical simulations, we verified that the derived lower bounds are indeed tight bounds, and the proposed bandwidth allocation algorithm can achieve a close-to-optimal peer upload bandwidth utilization.

³The bandwidth available for full video distribution is the total upload capacity minus the upload bandwidth utilized for transmitting voice and small videos.

We briefly describe the related work in Section II. The P2P relay framework for one-view MPVC is presented in Section III. In Section IV, we establish the universal video rate lower bound for homogeneous one-view MPVC. We study the video rate capacity region for heterogeneous MPVC in Section V. Two optimal P2P MPVC designs are studied to maximize the aggregate video quality and achieve the max-min fairness respectively. We also derive the *guaranteed max-min capacity* for heterogeneous one-view MPVC. In Section VI, we present a P2P relay bandwidth allocation algorithm to approach the maximal video rate region. In Section VII, we demonstrate the tightness of the derived lower bounds and the efficiency of the proposed bandwidth allocation algorithm through numerical simulations of randomly generated one-view MPVC scenarios. The paper is concluded with future work in Section VIII.

II. RELATED WORK

While P2P has been widely adopted for file sharing [3], [17] and video streaming [16], [10], only very limited efforts have been attempted for P2P MPVC in the research community. Chu *et al.* [5] proposed an End-System-Multicast architecture to support video conferencing applications, where multicast functionality is pushed to the edge. Lennox and Schulzrinne [12] proposed a full-mesh conferencing protocol without a central point of control. Luo *et al.* [14] proposed to integrate application layer multicast with native IP multicast in P2P conferencing systems. In [6], all users watching the same source form a chain and relay video to each other. Akkus *et al.* [1] extends this idea to relay video encoded in multiple layers. Recently, Chen *et al.* [4] proposed hybrid solutions to employ helpers to maximize the utility in P2P conferencing swarms, where helpers assist sources in relaying video streams to receivers. Ponc *et al.* [15] then extended this solution to support multi-rate conferencing applications with scalable coding techniques. None of the previous study investigate the impact of viewing relations on the achievable video capacity region. According to a recent study [9], a user on average watches one or two videos of other users since it is difficult for a user to simultaneously keep track of three or more video sources. It is often more preferable for a user to watch high quality videos of a couple of users of interests, rather than watch lousy videos of all users. Our work is motivated by the recent trend of one-view MPVC. We establish universal video rate lower bounds and propose P2P relay algorithm to achieve the maximal capacity region. Our study demonstrate that it is promising to develop a pure P2P solution for one-view MPVC.

III. P2P ONE-VIEW MPVC

A. One-view MPVC

We consider an one-view multi-party video conference, where, at any given time, each peer only watches one full video generated by another peer. We further assume a peer can switch among videos of other peers, the viewing relations between peers are time-varying. A snapshot of viewing relations among all peers in the conference is defined as an one-view

TABLE I
NOTATIONS

Notation	Definition
N	set of peers in the conferencing system
$S \subseteq N$	set of video-active peers (sources)
$I \triangleq N - S$	set of video-idle peers (pure viewers)
G_s	set of viewers of source $s \in S$
$G_s^{(S)} \triangleq G_s \cap S$	viewers of source s who are also sources
$G_s^{(I)} \triangleq G_s \cap I$	viewers of source s who are pure viewers
r_s	rate of video generated by source s
u_i	total upload bandwidth of peer i
$u_i^{(s)}, i \in S$	upload bandwidth of a source peer $i \in S$ allocated to the sub-conference it is hosting
$u_i^{(w)}$	upload bandwidth of peer i allocated to the sub-conference it is watching
$u_i^{(h)}$	upload bandwidth of peer i allocated to the common helper bandwidth pool
$B_s^{(W)}$	b.w. contributed to swarm s by its viewers
$B_s^{(H)}$	b.w. contributed to swarm s by helpers

MPVC scenario. As enumerated in Table I, the whole set of peers is denoted by N , with $n = |N|$ be the total number of peers. In a specific scenario, peers can be classified into two classes: the *video-active peers*, denoted by S , which are the peers being watched by some other peers, and the *video-idle peers*, denoted by I , which are the peers not watched by any other peer. We call each video-active peer $s \in S$ a video source, and use G_s to denote the subset of peers watching the video of s . We say peers in G_s participate in a sub-conference hosted by s . Since each peer watches exactly one video, $\{G_s, s \in S\}$ forms a partition of N , and we have $n = \sum_{s \in S} |G_s|$, where $|G_s|$ is the number of peers in sub-conference s . Since a peer watching s can also host her own sub-conference, we further partition the viewers of s into two subsets: $G_s^{(S)} \triangleq G_s \cap S$, the subset of viewers who are hosting their own sub-conferences; and $G_s^{(I)} \triangleq G_s \cap I$, the subset of viewers who are pure viewers.

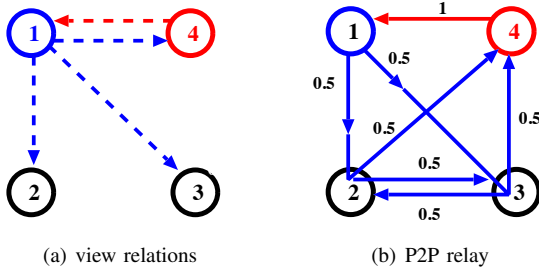


Fig. 1. One-view MPVC Example 1

Fig. 1(a) illustrates an one-view MPVC with four peers, with peer 2, 3 and 4 watching peer 1, and peer 1 watching peer 4. Therefore, we have $S = \{1, 4\}$, and $G_1 = \{2, 3, 4\}$, $G_1^{(S)} = \{4\}$, $G_1^{(I)} = \{2, 3\}$, $G_4 = G_4^{(S)} = \{1\}$, and $G_4^{(I)} = \emptyset$. If we assume peer upload bandwidth is the only bottleneck in the network and each peer has upload bandwidth of one unit, Fig. 1(b) plots one feasible bandwidth allocation scheme for this MPVC scenario: in swarm G_1 , peer 1 transfers different

video sub-streams at rate 0.5 to peer 2 and 3 individually, each of them then relays the received stream to the other two peers in G_1 ; in G_4 , peer 4 uploads its stream directly to peer 1 at rate 1. Under this bandwidth allocation scheme, all peers can receive video from their sources at rate 1. The idle peers will help the video source transfer the video stream and all peers' upload bandwidth is fully utilized.

Fig. 2(a) shows another watching scenario for the same MPVC: peer 2 and 3 watching peer 1; peer 1 watching peer 2; peer 4 watching peer 3. Fig. 2(b) plots one feasible bandwidth allocation scheme: for G_1 , peer 1 transfers one video sub-stream to peer 4 at rate 0.5, and peer 4 relays the received sub-stream to peer 2 and 3, peer 1 transfers another sub-stream at rate 0.25 directly to peer 2 and peer 3 with its remaining 0.5 bandwidth; for G_2 and G_3 , peer 2 and peer 3 upload their streams directly to their viewers at rate 1 respectively. Under this bandwidth allocation scheme, the received video rates on peer 1,2,3,4 are $(1, 3/4, 3/4, 1)$. In this bandwidth allocation scheme, even though peer 4 is not interested in video of peer 1, it helps peer 1 upload video with all its upload bandwidth. Fig. 2(c) shows another bandwidth allocation scheme which enables all peers' watching rate reach $7/8$. In this scheme, the video sources 2 and 3 reserve $1/8$ of bandwidth to relay the video it is watching.

The previous examples demonstrate that peer upload bandwidth allocation and peer perceived video quality depend on the viewing relations between peers. For general one-view MPVC, the first natural question to ask is: *given peer upload bandwidth profile, what are the maximal video source rates that can be supported under a specific viewing scenario?* It is expected that different viewing relations between peers will lead to different supportable video source rates. It is also tempting to ask the second question: *what are the maximal video source rates that can be supported under all possible viewing scenarios?* We will provide answers to these questions using analysis and simulations in the following sections.

B. P2P Video Relay

In this section, we formally introduce the P2P bandwidth sharing model in one-view MPVC. In P2P overlay networks, where each peer can reach all other peers, it is commonly assumed that peer upload links are the only bandwidth bottleneck [7], [4], [11]. In the rest of the paper, we adopt the assumption that the core network is congestion-free and video rates in MPVC are limited by peer's upload bandwidth.

To maximally utilize peer upload bandwidth, we assume all peers are fully cooperative. A peer not only can relay video that she is watching to other peers in the same sub-conference, she can also help peers watching a different source by downloading and relaying video of the source to which she has no interest to watch. Since video conferencing is highly delay-sensitive, to limit the delay incurred by relay, we also limit P2P video relay to two overlay hops, i.e., video can be relayed by at most one intermediate peer from a source to all

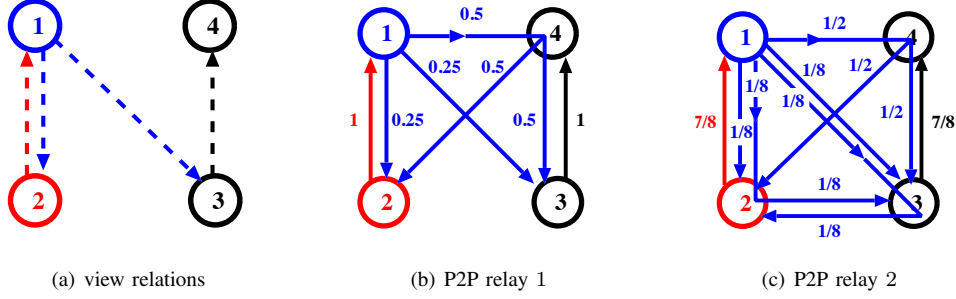


Fig. 2. One-view MPVC Example 2: video rates determined by P2P relay

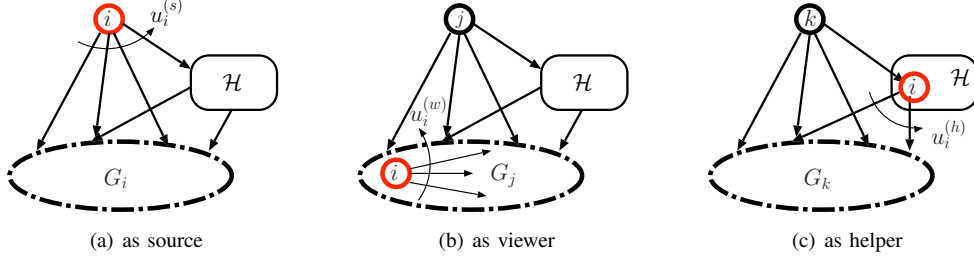


Fig. 3. Different Roles of a Peer in MPVC.

its receivers.⁴

Fig. 3 illustrates the concept of P2P video relay among peers in different sub-conferences. Let's first focus on a source peer $i \in S$. Without loss of generality, peer i is hosting a sub-conference G_i , while watching the video of another source, say peer j . In Fig. 3(a), peer i divides its video into multiple sub-streams, with possibly unequal rates. Then it sends these sub-streams to peers in its own sub-conference G_i . Each peer is responsible for duplicating and relaying the received sub-stream to all other peers in G_i . Peers outside of G_i can also help redistribute i 's video. We call those peers *the helpers* of G_i . Peer i sends a sub-stream to a helper, who then relays the sub-stream back to peers in G_i . In Fig. 3(b), other than distributing its own video, peer i , as a viewer in sub-conference G_j , is also responsible for redistributing the video of peer j . Additionally, peer i may also act as a helper to help the sub-conferences that she is not hosting, nor watching. In Fig. 3(c), peer i helps relay the video of peer k even though peer i does not watch video of k . Let's now examine how a source peer $i \in S$ allocates its upload bandwidth of u_i among its three roles in MPVC.

- 1) As a video source, peer i allocates $u_i^{(s)}$ bandwidth to upload her own video. $u_i^{(s)}$ consists of the bandwidth used to upload video directly to her viewers and the bandwidth used to upload video to her helpers;
- 2) As a viewer, peer i allocates $u_i^{(w)}$ bandwidth to relay the video of the source she is watching;
- 3) As a helper, peer i allocates $u_i^{(h)}$ bandwidth to relay the video of other sources that she is not watching.

⁴It has also been shown that two-hop relay is bandwidth optimal in uplink throttled P2P systems [4], [11]

For a peer not hosting a sub-conference, i.e., $i \in I$, she has dual roles: viewer and helper. She only needs to split her upload bandwidth between $u_i^{(w)}$ and $u_i^{(h)}$.

Given the bandwidth allocation on all peers, we can calculate the bandwidth resource available to each sub-conference. For the sub-conference hosted by peer i , there are three portions of bandwidth available. The first portion is the bandwidth contributed by peer i itself and it is $u_i^{(s)}$. The second portion is the bandwidth contributed by its viewers $B_i^{(w)} \triangleq \sum_{j \in G_i} u_j^{(w)}$. The third portion of the available bandwidth is contributed by all helpers of sub-conference. In principle, any peer not in G_i can be a helper of G_i . Instead of tracking bandwidth allocation of each helper to each sub-conference, we build a common helper pool \mathcal{H} to manage bandwidth contributed by all helpers. More specifically, each peer $i \in N$ contributes $u_i^{(h)}$ amount of bandwidth to the helper pool \mathcal{H} . The total bandwidth available in \mathcal{H} is therefore $B^{(H)} \triangleq \sum_{i \in N} u_i^{(h)}$. The manager of the helper pool is in charge of distributing $B^{(H)}$ to different sub-conferences. Let $B_s^{(H)}$ be the helper bandwidth allocated to sub-conference s , then we have $\sum_{s \in S} B_s^{(H)} \leq \sum_{i \in N} u_i^{(h)}$. In the following, we treat the helper bandwidth allocated to a sub-conference s as if it is from a single virtual helper with total upload bandwidth of $B_s^{(H)}$. As will be shown shortly, such a centralized bandwidth management and helper virtualization can achieve the maximal video rates in MPVC.

Peer upload bandwidth allocation $\mathcal{U} \triangleq \{u_i^{(s)}, u_i^{(w)}, u_i^{(h)}, i \in N\}$ determines the upload bandwidth available for each source to distribute her video to her viewers. In the sub-conference hosted by source $i \in S$, as shown in [13], [4], the maximal

achievable video rate is:

$$r_i^* = \min \left\{ u_i^{(s)}, \frac{u_i^{(s)} + B_i^{(W)} + B_i^{(H)}}{|G_i|} - \frac{B_i^{(H)}}{|G_i|^2} \right\}, \quad (1)$$

where $B_i^{(W)}$ is bandwidth contributed by the viewers, and $B_i^{(H)}$ is bandwidth borrowed from the helper pool.

IV. CAPACITY OF HOMOGENEOUS MPVC

Equation (1) states that the achievable video rate in each sub-conference is determined by peer upload bandwidth allocation. In the following two sections, we will study the optimal peer bandwidth allocation to achieve high video rates cross all sub-conferences. In particular, we establish several non-trivial video rate lower bounds *independent of the viewing relations between peers*.

In this section, we assume peers are homogeneous and have normalized upload bandwidth of 1. The simplest bandwidth assignment is to assign peer bandwidth to each sub-conference at the gratuity of 1. This means that a video source will use all its bandwidth to transfer its own video, i.e., $u_s^{(s)}=1$, $u_s^{(w)} = u_s^{(h)} = 0$, $\forall s \in S$. An idle peer $i \in I$ utilizes its bandwidth either to transfer the stream it is watching or to help other sub-conferences, i.e., $u_i^{(w)} + u_i^{(h)} = 1$, and $u_i^{(w)} * u_i^{(h)} = 0$, $\forall i \in S$. The bandwidth allocation becomes an idle peer assignment problem: *how to assign idle peers to sub-conferences to maximize their video rates?*

Theorem 1: For homogeneous one-view MPVC with two sources, both sources can achieve the maximum rate of 1.

Proof: Without loss of generality, suppose source 1 has $|G_1|$ viewers, and source 2 has $|G_2|$ viewers. Since each source always watches another source, we must have the two sources watch each other, i.e., $1 \in G_2$ and $2 \in G_1$. Then there are $|G_1| - 1$ and $|G_2| - 1$ idle peers in sub-conference 1 and 2 respectively. If we let each idle peer only relay video she is watching, then we have $u_1^{(s)} = u_2^{(s)} = 1$, $B_1^{(W)} = |G_1| - 1$, $B_2^{(W)} = |G_2| - 1$, and $B_1^{(H)} = B_2^{(H)} = 0$. According to (1), we have $r_1^* = r_2^* = 1$. ■

From the proof of Theorem 1, we know that, to achieve high video rates, it is important to have enough idle peers to upload in each sub-conference. For general cases with more than two sources, we have the following result.

Lemma 1: For any one-view MPVC scenario, we have

$$\sum_{i \in S} |G_i^{(I)}| = \sum_{i \in S} (|G_i| - 1) \quad (2)$$

Proof: Since a peer is either an idle peer or a busy peer, thus $|N| = |S| + |I|$. In addition, every peer watches exactly one video. Hence $|N| = \sum_{i \in S} |G_i|$. So we have

$$|I| = \sum_{i \in S} |G_i| - |S| = \sum_{i \in S} (|G_i| - 1)$$

In addition, since $\{G_i, i \in S\}$ is a partition of N , and $G_i^{(I)} = G_i \cap I$, then $\{G_i^{(I)}, i \in S\}$ is a partition of $I \subset N$. Then we have $\sum_{i \in S} |G_i^{(I)}| = |I| = \sum_{i \in S} (|G_i| - 1)$ ■

Base on Lemma 1, we now present an idle peer assignment procedure that can guarantees each sub-conference with G_i users can be assigned with $|G_i| - 1$ idle peers.

- 1) For a sub-conference where all viewers are idle, i.e. $G_i^{(I)} = G_i$, it will only use $|G_i| - 1$ of its own viewers to relay video and $B_i^{(H)} = 0$, it also contributes one idle peer to the common helper pool \mathcal{H} ;
- 2) For a sub-conference where exactly one viewer is a source, i.e. $|G_i^{(I)}| = |G_i| - 1$, it will only use all of its own viewers to relay video and $B_i^{(H)} = 0$, it does not contribute any peer to the helper pool \mathcal{H} ;
- 3) For a sub-conference with $|G_i^{(I)}| < |G_i| - 1$, it will use its own $|G_i^{(I)}|$ idle peers and $|G_i| - 1 - |G_i^{(I)}|$ idle peers from the helper pool \mathcal{H} to relay its video, i.e., $B_i^{(H)} = |G_i| - 1 - |G_i^{(I)}|$.

If we use S_1 , S_2 and S_3 to represent the set of sub-conferences in case 1), 2), and 3) respectively, then the number of helpers contributed to \mathcal{H} by sub-conferences in S_1 is:

$$\begin{aligned} |\mathcal{H}| &= \sum_{i \in S_1} |G_i^{(I)}| - (|G_i| - 1) \\ &= \sum_{k=2,3} \sum_{i \in S_k} |G_i| - 1 - |G_i^{(I)}| \\ &= \sum_{i \in S_3} |G_i| - 1 - |G_i^{(I)}| = \sum_{i \in S_3} B_i^{(H)}, \end{aligned}$$

where the second equality is due to Lemma 1 and the third equality is due to $|G_i^{(I)}| = |G_i| - 1$ for any sub-conference in S_2 . This guarantees that the previous helper allocation scheme is feasible. Idle peers assigned to sub-conference of source i include its own idle viewers in $G_i^{(I)}$, and idle peers from other sub-conferences.

Theorem 2: If all peers' bandwidth is one, for any given scenario $\{G_i, i \in S\}$, the achievable video rate r_i for any sub-conference G_i satisfy:

$$r_i = 1 - \frac{B_i^{(H)}}{|G_i|^2} \geq 1 - \frac{1}{|G_i|} + \frac{1}{|G_i|^2} \geq \frac{3}{4} \quad (3)$$

Proof: In the previous idle peer assignment,

$$u_i^{(s)} = 1, \quad B_i^{(W)} + B_i^{(H)} = |G_i| - 1, \forall i \in S. \quad (4)$$

According to (1), the achievable rate is

$$r_i = \frac{1 + |G_i| - 1}{|G_i|} - \frac{B_i^{(H)}}{|G_i|^2} = 1 - \frac{B_i^{(H)}}{|G_i|^2}.$$

Since $B_i^{(H)} \leq |G_i| - 1$, we have

$$r_i \geq 1 - \frac{1}{|G_i|} + \frac{1}{|G_i|^2}$$

Let $f(x) = 1 - x^{-1} + x^{-2}$, $f(x)$ is an increasing function when $x \geq 2$, and $f(2) = \frac{3}{4}$; Thus, when $|G_i| \geq 2$, $r_i \geq \frac{3}{4}$, when $|G_i| = 1$, the source send its stream directly to the only viewer, and $r_i = 1$. ■

Theorem 2 applies to any one-view MPVC. *The lower bound of 3/4 is independent of the viewing relations between*

peers. This non-trivial lower bound has important implications on the practical implementation of MPVC, within which a peer may join or leave a sub-conference at her will. It is undesirable to change the video rates of sub-conferences frequently whenever the viewing relations change. Our results suggest that it is possible to find a constant rate for all video sources that is achievable in any possible one-view MPVC scenario, independent of the viewing relations among peers, sub-conference sizes, and even the total number of peers in the system. We name the maximum value of such a constant source rate as the *guaranteed capacity* of one-view MPVC and denote this value as C_1 .

Theorem 3: Under the assumption that all peers have homogeneous upload bandwidth of 1, the guaranteed capacity C_1 for any homogeneous one-view MPVC is $5/6$.

Proof: In the configuration of Theorem 2, the video source uses up its upload bandwidth to distribute the video stream to other peers in its sub-conference. Here we will use a slightly different video distribution configuration to achieve a higher bound of the capacity C_1 . In this configuration, all source peer will use rate w to upload the its own video while the remaining upload bandwidth of $1 - w$ is used to distribution the video it is watching. On the other hand, idle peers are still assigned to different sub-conferences in the same way as in Theorem 2. An idle peer will contribute its full upload bandwidth to help transmitting the video assigned to it. Under this configuration, besides the helper bandwidth, the busy peers in G_i also contribute upload bandwidth to sub-conference i . According to Equation (1),

$$v_i(w) = \frac{w + (1-w)|G_i^{(S)}| + |G_i| - 1}{|G_i|} - \frac{B_i^{(H)}}{|G_i|^2} \quad (5)$$

Case 1: If all viewers of source i are idle peers, and $|G_i^{(S)}| = 0$. According to the idle peer assignment rule, $B_i^{(H)} = 0$, $\forall i \in S_1$. Equation (5) becomes

$$v_i(w) = \frac{w + |G_i| - 1}{|G_i|}$$

To have $v_i(w) \geq w$, we need

$$v_i(w) = \frac{w + |G_i| - 1}{|G_i|} \geq w \Rightarrow |G_i| - 1 \geq (|G_i| - 1)w \Rightarrow 1 \geq w$$

Since $w < 1$, we always have $v_i(w) \geq w$. Thus $r_i = w$.

Case 2: If exactly one viewer of source i is a source, and $|G_i^{(S)}| = 1$. In this case, $B_i^{(H)} = 0$. Equation (5) becomes

$$v_i(w) = \frac{w + (1-w) + |G_i| - 1}{|G_i|} = 1$$

Thus, Thus $r_i = w$ in this case.

Case 3: If more than one viewer of source i are sources, and $|G_i^{(S)}| \geq 2$. Substitute $|G_i^{(S)}| = |G_i| - |G_i^{(I)}|$ and $B_i^{(H)} = |G_i| - 1 - |G_i^{(I)}|$ into Equation (5), we have

$$v_i(w) = (2-w)\left(1 - \frac{1}{|G_i|}\right) + \frac{1}{|G_i|^2} + (1-w)\frac{|G_i| + 1}{|G_i|^2}|G_i^{(I)}|$$

Since $|G_i| \geq |G_i^{(S)}| \geq 2$, if we set $w = \frac{5}{6}$, when $|G_i| \geq 4$, we have

$$v_i(5/6) > (2 - 5/6)\left(1 - \frac{1}{|G_i|}\right) \geq (2 - 5/6)(1 - 1/4) = 7/8.$$

when $|G_i| = 3$, we have

$$v_i(5/6) > (2 - 5/6)(1 - 1/3) + 1/9 \geq 8/9 > \frac{5}{6}$$

when $|G_i| = 2$, we have

$$v_i(5/6) > (2 - 5/6)(1 - 1/2) + 1/4 \geq 5/6$$

In all cases, we will have $r_i \geq 5/6$, therefore we conclude that $C_1 \geq 5/6$.

Finally, to show this $5/6$ is a tight bound of the guaranteed capacity, we only need to come up with a homogeneous MPVC scenario such that the maximal achievable rate on all video sources is only $5/6$. Since we will use an optimization formulation for the more general heterogeneous MPVC scenario, we present it as a constructive proof in Appendix A \blacksquare

V. CAPACITY OF HETEROGENEOUS MPVC

In the previous section, we assume peer upload bandwidth is homogeneous and only assign idle peers to different sub-conferences. In practice, peer upload bandwidth is heterogeneous. Peer upload bandwidth should be allocated to sub-conferences at finer granularity than 1. In this section, we study optimal peer bandwidth allocation schemes to achieve different design objectives in heterogeneous MPVC systems.

A. Maximizing Aggregate Video Quality

The first design objective is to maximize the total video quality received by all peers. We adopt a PSNR-type of video quality model [4], which quantifies the quality of a video stream at rate r_i as $\log(r_i)$. The optimal peer bandwidth allocation is to maximize the total video quality of the conference:

$$\text{OPT I: } \max_{\mathcal{U}, \mathcal{R}, \mathcal{B}} \sum_{i \in S} |G_i| \log(r_i), \quad (6)$$

subject to :

$$r_i \leq \frac{u_i^{(s)} + \sum_{j \in G_i} u_j^{(w)} + B_i^{(H)}}{|G_i|} - \frac{B_i^{(H)}}{|G_i|^2}, \quad (7)$$

$$r_i \leq u_i^{(s)}, \quad \forall i \in S \quad (8)$$

$$u_i \geq u_i^{(s)} + u_i^{(w)} + u_i^{(h)}, \quad \forall i \in S \quad (9)$$

$$u_i \geq u_i^{(w)} + u_i^{(h)}, \quad \forall i \in I \quad (10)$$

$$\sum_{s \in S} B_s^{(H)} \leq \sum_{i \in N} u_i^{(h)}, \quad (11)$$

where (7) and (8) are source video rate constraints according to Equation (1), (9) and (10) are upload bandwidth constraints on sources and idle peers respectively, and (11) enforces the bandwidth supply and demand balance in the common helper pool. The objective function is a concave function of $\{r_i\}$ and the constraints are all linear. It is a convex optimization

problem, for which efficient centralized and distributed algorithms can be developed to solve for the optimal video source rates $\mathcal{R}^* = \{r_i^*, i \in S\}$ and the associated optimal P2P relay scheme characterized by the peer upload bandwidth allocation $\mathcal{U}^* = \{u_i^{(s)*}, u_i^{(w)*}, u_i^{(h)*}, i \in N\}$ and helper bandwidth allocation $\mathcal{B}^* = \{B_s^{(H)*}, s \in S\}$.

Due to the log video utility function, the optimal solution of OPT I achieves the *weighted proportional fairness* among all video sources, with the weight for a sub-conference be the number of viewers.

B. Achieving Max-Min Fairness

Another widely used fairness metric is the *max-min fairness*. Intuitively, we prefer all sources to achieve the same rate as long as it allows by the individual source's upload capacity and the available bandwidth resource in the whole MPVC system. To achieve this, we want to find a video rate γ such that if a video source i 's upload capacity u_i is less than γ , it should be able to stream its video at rate $r_i = u_i$, for any other source with $u_i \geq \gamma$, it should stream its video at the common rate $r_i = \gamma$. Under this setting, the capacity of the system is defined as the maximal supportable γ , which can be solved by the following optimization problem.

$$\text{OPT II: } \max_{\mathcal{U}, \mathcal{R}, \mathcal{B}} \gamma \quad (12)$$

subject to (7), (8), (9), (10), (11) and a new set of constraints

$$r_i = \min(\gamma, u_i), \quad \forall i \in S \quad (13)$$

OPT II is no longer a simple convex programming problem due to the non-linear constraints in (13). To solve it, we divide the solution space of γ according to the bandwidth distribution of all video sources. Specifically, we first sort the upload bandwidth of video sources in a non-decreasing order and denote the sequence as $\{b_1, b_2, \dots, b_{|S|}\}$, with $b_i \leq b_j, \forall i < j$. We then condense the list into a strictly increasing list $\{c_1, c_2, \dots, c_m\}$ by removing redundant values. Let $c_0 = 0$, and $c_{m+1} = \infty$, the solution space for γ can be divided into $m + 1$ intervals: $[c_k, c_{k+1}), 0 \leq k \leq m$. When casted into interval k , OPT II becomes a linear programming problem with (13) being replaced by a set of linear constraints:

$$r_i = \begin{cases} u_i, & \forall i \in S \text{ such that } u_i \leq c_k \\ \gamma, & \forall i \in S \text{ such that } u_i > c_k \end{cases} \quad (14)$$

We can solve OPT II iteratively, starting from interval 0 until the first interval k_0 where the optimal solution of OPT II satisfies $\gamma_{k_0}^* < c_{k_0+1}$. Then $\gamma_{k_0}^*$ is the final solution of OPT II: $\gamma^* = \gamma_{k_0}^*$, and the optimal source rates are $r_i^* = u_i$ if $u_i \leq \gamma^*$ and $r_i^* = \gamma^*$ if $u_i > \gamma^*$.

Theorem 4: The optimal source rates obtained in solving OPT II is max-min fair.

Proof: According to the definition of max-min fairness, an allocation vector X is max-min fair if and only if X , when sorted in non-decreasing order, is lexicographically maximal among all feasible allocation vectors sorted in non-decreasing

order. We prove the theorem using contradiction argument. Let's assume the optimal source rates \mathcal{R}^* of OPT II is not max-min fair, then there must exist another source rate vector \mathcal{R}^0 which is lexicographically larger than \mathcal{R}^* . In other words, if we sort both \mathcal{R}^* and \mathcal{R}^0 into non-decreasing order, there exists an index k such that $r_i^* = r_i^0$ for $i = 1, \dots, k$, and $r_{k+1}^* < r_{k+1}^0$. Without loss of generality, we sort peer id in non-decreasing order of their upload capacity, let w be the peer id such that $u_w < \gamma^*$ and $u_{w+1} \geq \gamma^*$, then we know that $\mathcal{R}^* = \{u_1, \dots, u_w, \gamma^*, \dots, \gamma^*\}$. For any peer i , $1 \leq i \leq w$, its video rate is constrained by its own upload capacity. In any other feasible solution, including \mathcal{R}^0 , the highest possible video rate for peer i is still u_i . In other words, the first w components of \mathcal{R}^0 (when sorted in non-decreasing order) is upper bounded by $\{u_1, \dots, u_w\}$, with component-wise vector comparison. So we must have $k \geq w$. If $k = w$, then $r_{w+1}^0 > \gamma^*$, and a new vector $\mathcal{R}^1 \triangleq \{u_1, \dots, u_w, r_{w+1}^0, \dots, r_{w+1}^0\}$ is a feasible source rate vector, and $r_{w+1}^0 > \gamma^*$ is a better solution for OPT II. This contradicts with the fact that γ^* is the optimal solution of OPT II. If $k > w$, then $\mathcal{R}^0 = \{u_1, \dots, u_w, \gamma^*, \dots, \gamma^*, r_{k+1}^0, \dots\}$. Then for source peer $k+1$, we can reduce its video rate by an amount of $\Delta = \frac{r_{k+1}^0 - \gamma^*}{2}$, and contribute the saved upload bandwidth Δ on peer $k+1$ to the helper pool to increase the rate of source $w+1$ through k . Specifically, let

$$\epsilon = \min \left\{ u_{w+1} - \gamma^*, \frac{\Delta}{\sum_{i=w+1}^k |G_i|} \right\}.$$

By allocating $\epsilon |G_i|$ helper bandwidth to sub-conference i , with $w+1 \leq i \leq k$, we increase the video rates of sub-conferences from $w+1$ to k by ϵ . Then the newly achieved video rates vector is $\mathcal{R}^2 = \{u_1, \dots, u_w, \gamma^* + \epsilon, \dots, \gamma^* + \epsilon, r_{k+1}^0 - \Delta, \dots\}$. Consequently, $\gamma^* + \epsilon$ is a better solution of OPT II than γ^* . This again contradicts with the fact that γ^* is the optimal solution of OPT II.

In conclusion, there is no feasible video source rate vector which is lexicographically larger than \mathcal{R}^ . The optimal source rates obtained in solving OPT II is max-min fair.* ■

Definition Max-min Capacity: we define the optimal solution of OPT II γ^* as the max-min capacity of a heterogeneous one-view MPVC scenario.

C. Lower Bound of Max-min Capacity

While the max-min capacity γ^* for each one-view MPVC scenario can be iteratively solved for the corresponding optimization problem OPT II, similar to the homogeneous case, it is important to obtain lower bounds of γ^* for heterogeneous systems that is independent of specific watching relations, and even better, independent of conference sizes.

We normalize peers' upload bandwidth such that the average peer bandwidth is unit one. Then we have $\sum_{i \in N} u_i = |N|$. We first establish a lower bound for γ^* as a function of the number of sources and the number of peers in MPVC, but independent of the viewing relations among peers.

Theorem 5: For any one-view MPVC with $|N|$ peers and $|S|$ sources, for any viewing scenario, we have

$$\gamma^* \geq \frac{|N|}{|N| + |S|}$$

Proof: We prove it by constructing a peer and helper bandwidth allocation scheme that leads to $\gamma_0 \triangleq \frac{|N|}{|N| + |S|}$. Specifically, for each source peer $i \in S$, its video rate is $r_i = \min(u_i, \gamma_0)$, and its bandwidth allocation scheme is $u_i^{(s)} = r_i$, $u_i^{(w)} = 0$, $u_i^{(h)} = u_i - r_i$, i.e., each source peer only reserves upload bandwidth of its own source rate r_i , and contributes the remaining bandwidth to the helper pool. For each idle peer $i \in I$, the bandwidth allocation scheme is $u_i^{(w)} = 0$, $u_i^{(h)} = u_i$, i.e., each idle peer contributes all its upload bandwidth to the common helper pool. Under such a bandwidth allocation, source i first uploads its video to the helper pool using its reserved upload bandwidth $u_i^{(s)} = r_i$, then the helpers will duplicate and relay a copy to each peer in G_i . The total helper bandwidth needed by sub-conference i is $B_i^{(H)} = |G_i|r_i$.

To make the bandwidth allocation scheme feasible, the demand of helper bandwidth should be less than the supply of helper bandwidth, that is

$$\sum_{i \in S} r_i |G_i| \leq \sum_{i \in N} u_i - \sum_{j \in S} r_j.$$

That is

$$\sum_{i \in S} (r_i |G_i| + r_i) \leq \sum_{i \in N} u_i = |N| \quad (15)$$

Since $r_i \leq \gamma^0 = \frac{|N|}{|N| + |S|}$, the left-hand side of (15)

$$\begin{aligned} \sum_{i \in S} r_i (|G_i| + 1) &\leq \frac{|N|}{|N| + |S|} \sum_{i \in S} (|G_i| + 1) \\ &= \frac{|N|}{|N| + |S|} (|N| + |S|) = |N| \end{aligned}$$

Thus, γ^0 is a feasible solution of OPT II and $\gamma^* \geq \frac{|N|}{|N| + |S|}$. ■

While the previous lower bound depends on the number of sources and viewers, it is still desirable to establish lower bounds of γ^* which applies to any one-view MPVC. We call the maximum of such lower bounds the *guaranteed max-min capacity* of one-view MPVC.

Theorem 6: The guaranteed max-min capacity of one-view MPVC is $2/3$.

Proof: We first prove $2/3$ is a guaranteed lower bound by constructing a specific bandwidth allocation scheme to achieve $\gamma = 2/3$ in any one-view MPVC.

Firstly, we allocate peer bandwidth as follows:

$$u_i^{(s)} = \min(u_i, 1), u_i^{(w)} = 0, u_i^{(h)} = u_i - u_i^{(s)}; \forall i \in S \quad (16)$$

$$u_i^{(w)} = 0, u_i^{(h)} = u_i, \forall i \in I \quad (17)$$

In (16), a video source with bandwidth larger than one reserves bandwidth of one to transfer its own video, and contributes the remaining bandwidth to the helper pool; a video source with bandwidth less than one uses up all its bandwidth to transfer its own video. In (17), idle peers contribute all their bandwidth to the common helper pool.

Secondly, we assign the helper bandwidth to each sub-conference as $B_s^{(H)} = |G_s| - u_i^{(s)}$, $\forall s \in S$. The total helper bandwidth needed is

$$\begin{aligned} \sum_{s \in S} B_s^{(H)} &= \sum_{s \in S} |G_s| - \sum_{s \in S} u_i^{(s)} = |N| - \sum_{s \in S} u_i^{(s)} \\ &= \sum_{i \in N} u_i - \sum_{s \in S} u_i^{(s)} = \sum_{i \in S} (u_i - u_i^{(s)}) + \sum_{i \in I} u_i \\ &= \sum_{i \in N} u_i^{(h)}, \end{aligned}$$

where the second equality is due to the total number of reviewers is $|N|$, the third equality is due to the total user upload bandwidth (after normalization) is $|N|$, the last equality is due to the previous upload bandwidth allocation on sources and idle peers. The sequence of equalities show that the total helper bandwidth needed equals to the total helper bandwidth contributed. This bandwidth allocation scheme is feasible.

Now we calculate the achieved video rates under this peer and helper bandwidth allocation scheme.

For a source i with $u_i > 1$, according to Equation (1), we have

$$r_i = \frac{1 + |G_i| - 1}{|G_i|} - \frac{|G_i| - 1}{|G_i|^2} = 1 - \frac{|G_i| - 1}{|G_i|^2}$$

As discussed in section IV, $r_i \geq 3/4$.

For a source i with $u_i \leq 1$, according to Equation (1), we have

$$v_i = \frac{u_i + |G_i| - u_i}{|G_i|} - \frac{(|G_i| - u_i)}{|G_i|^2} = 1 - \frac{|G_i| - u_i}{|G_i|^2} \quad (18)$$

If $|G_i| = 1$, $v_i = u_i$.

If $|G_i| \geq 2$, let $f(x) = 1 - x^{-1} + u_i x^{-2}$, then $\frac{df(x)}{dx} = x^{-2} - 2u_i x^{-3} = x^{-3}(x - 2u_i)$. $f(x)$ is an increasing function when $x \geq 2u_i$. For any given $u_i \leq 1$, v_i is an increasing function of $|G_i|$ when $|G_i| \geq 2$, and the minimal value is $1/2 + u_i/4$ when $|G_i| = 2$. For a source with upload capacity $2/3 \leq u_i \leq 1$, $v_i \geq 1/2 + \frac{2}{3}/4 = 2/3$, so the achieved video rate $r_i = \min(u_i, v_i) \geq 2/3$. Finally, for a source with $u_i < 2/3$, we automatically have $u_i < 1/2 + u_i/4 \leq v_i$, the achieved video rate $r_i = \min(u_i, v_i) = u_i$, i.e., the video rate is constrained by the source upload capacity.

In conclusion, with the proposed bandwidth allocation scheme, the achieved video source rates are:

$$\begin{aligned} r_i &= u_i, \text{ if } u_i < \frac{2}{3}; \\ r_i &\geq \frac{2}{3}, \text{ if } \frac{2}{3} \leq u_i < 1; \\ r_i &\geq \frac{3}{4}, \text{ if } u_i \geq 1. \end{aligned}$$

Thus, $\gamma = \frac{2}{3}$ is a lower bound of γ^* for any one-view MPVC. The guaranteed max-min capacity is at least $2/3$.

Now we prove the guaranteed max-min capacity cannot be higher than $2/3$ by constructing one-view MPVC with $\gamma^* \rightarrow 2/3$.

We construct the following one-view MPVC: there are $2m+1$ peers (m is a positive integer). Among these peers, there are m peers acting as video sources, and one source peer has bandwidth of $1 + \epsilon$, with ϵ be a small positive value, each other video source's bandwidth is one. In addition, there is a super peer whose bandwidth is $2m(1 - \epsilon)$. The remaining m peers' upload bandwidth is zero. Each source watches video of another source, and no two sources watch the same source. Each idle peer watches one source, and no two idle peers watch the same source. Finally, the super peer watches the source with upload bandwidth of $1 + \epsilon/2$. All sources with bandwidth one has two viewers, the source with bandwidth of $1 + \epsilon/2$ has three viewers. The bandwidth allocation scheme to maximize γ is: each video source uploads its video to the super peer at rate $1 - \epsilon$ and the super peer relays it to each of the two viewers in the sub-conference. Each video source also uploads directly to each of its viewer at rate $\epsilon/2$. The achieved video rate of each sub-conference is $1 - \epsilon/2$, which is less than each source's upload capacity. The average upload bandwidth of the conference is $\frac{2m(1-\epsilon)+m+\epsilon/2}{2m+1}$. Then the max-min capacity γ^* is the achieved video rate normalized against the average upload bandwidth

$$\gamma^*(m, \epsilon) = \frac{(2m+1)(1-\epsilon/2)}{2m(1-\epsilon)+m+\epsilon/2}$$

Since $\lim_{m \rightarrow \infty, \epsilon \rightarrow 0} \gamma^*(m, \epsilon) = 2/3$, we can construct a sequence of one-view MPVCs with max-min fairness capacity approaching $2/3$ from the above. So the guaranteed max-min fairness capacity for arbitrary one-view MPVC is $2/3$. ■

With Theorem 5, and Theorem 6, we immediately have

Corollary 1: For any one-view MPVC with $|N|$ peers and $|S|$ sources, under any viewing scenario, we have

$$\gamma^* \geq \max \left\{ \frac{|N|}{|N| + |S|}, \frac{2}{3} \right\}$$

When proving theorem 6, we noticed that for a source with bandwidth greater than 1, its video rate can be larger than $3/4$. This suggests that if all sources have upload bandwidth above the average upload bandwidth, the max-min capacity can be made large.

Corollary 2: If the bandwidth of all video sources is larger than one, we have $\gamma^* \geq 3/4$.

Proof: Similar to the proof in Theorem 6, we assume each video source allocates bandwidth of one in its own sub-conference, and contributes the remaining bandwidth to the helper pool. The bandwidth of each idle peer is contributed to the helper pool. The total bandwidth in the helper pool is $|N| - |S|$.

Since $\sum_{i \in S} (|G_i| - 1) = \sum_{i \in S} |G_i| - \sum_{i \in S} 1 = |N| - |S|$. Thus each sub-conference can be assigned with $|G_i| - 1$ helper

bandwidth, i.e., $B_i^{(H)} = |G_i| - 1$. From (1), we have

$$r_i = \frac{1 + |G_i| - 1}{|G_i|} - \frac{|G_i| - 1}{|G_i|^2} = 1 - \frac{|G_i| - 1}{|G_i|^2}$$

As discussed in section IV, $r_i \geq 3/4, \forall i \in S$. So we have $\gamma^* \geq 3/4$. ■

VI. P2P MPVC RELAY DESIGN

In the previous two sections, we characterized the video rate capacity region for one-view MPVC. In this section, we propose peer bandwidth allocation algorithms to achieve a feasible video source rate vector within the capacity region. Instead of squeezing all peers' upload bandwidth to achieve the maximal video rates, we focus on supporting a given video rate vector with the minimum peer upload bandwidth through efficient bandwidth allocation. The saved peer bandwidth provides a cushion to absorb the impacts of peer churn and network bandwidth variations incurred in practical MPVC systems.

We first present bandwidth allocation guidelines obtained from the capacity region study. We then develop peer bandwidth allocation algorithm to implement a given video source rate vector. Finally, we show the bandwidth allocation algorithm can also be used to explore the capacity region through binary search.

A. Design Guidelines

As discussed in Section III-B, a peer can play three different roles in a sub-conference: source, viewer, and helper. A peer allocates its upload bandwidth among the three different roles. To develop efficient bandwidth allocation algorithm, let's first examine how different roles contribute to the achieved video rate. According to (1), if source i is not constrained by its own upload capacity, the achieved video rate can be rewritten as

$$r_i^* = \frac{w_i^{(s)}}{|G_i|} + \frac{B_i^{(W)}}{|G_i|} + \frac{B_i^{(H)}}{|G_i|} \left(1 - \frac{1}{|G_i|} \right), \quad (19)$$

where a unit bandwidth from either the source or a viewer increases the video rate by $1/|G_i|$, but the contribution of a unit helper bandwidth is discounted by a factor of $(1 - 1/|G_i|)$. The discount reflects the overhead of employing a helper. Specifically, whenever swarm i employs a helper, source i has to first stream some video to the helper so that it can relay video back to the viewers of swarm i . Since the helper itself is not a viewer in swarm i , the bandwidth used to stream video to it does not directly contribute to the achieved video rate in swarm i . The overhead is inversely proportional to $|G_i|$ and decreases with the size of the sub-conference being helped. An efficient bandwidth allocation should maximally avoid helper bandwidth overhead. This leads to the first guideline:

G1: A sub-conference should maximally utilize bandwidth available on its source and viewers before using helpers. A peer should always allocate its bandwidth to the sub-conference she is hosting or viewing before contributing bandwidth to the helper pool.

To avoid helper bandwidth overhead, an idle viewer's bandwidth can only be used by the sub-conference she is viewing,

but the bandwidth of a video source can be utilized by two sub-conferences: the sub-conference that she is hosting and the sub-conference that she is viewing. To preserve bandwidth allocation flexibility, we propose the second guideline:

G2: A sub-conference i with target video rate r_i first draws bandwidth r_i from its source, then it should maximally utilize bandwidth available on its idle viewers before drawing additional bandwidth from its source and busy viewers.

From (19), the helper bandwidth overhead is a decreasing function of the sub-conference size. Between the two sub-conferences that a video source can upload to without overhead, the one with the smaller number of viewers would incur higher helper bandwidth overhead if it uses bandwidth from the helper pool. To reduce the system-wide helper bandwidth overhead, we have the third guideline:

G3: If a source has surplus bandwidth over its target video rate, between the two sub-conferences that she is hosting and viewing, she should allocate the surplus bandwidth first to the sub-conference with the smaller number of viewers.

B. Bandwidth Allocation Algorithm

Now we present our bandwidth allocation algorithm based on the three guidelines. We adopt two-level hierarchy for bandwidth management. At the top level, a centralized tracker manages the helper pool shared by all sub-conferences. It keeps track of the bandwidth contributed by peers in sub-conferences with surplus bandwidth, and allocates helper bandwidth to sub-conferences with bandwidth deficit. At the bottom level, the bandwidth allocation among peers in each sub-conference is coordinated by the video source. Source of sub-conference i maintains the following states:

- 1) r_i : the target video rate for sub-conference i ;
- 2) $B_i^{(H)}$: the helper bandwidth borrowed from the helper pool, initialized to 0.
- 3) A_i : achieved video rate under current allocation, initialized to 0.
- 4) L_j : bandwidth on peer $j \in G_i$ that has not been allocated, initialized to u_j .

Bandwidth allocation is carried out in four stages: video source bandwidth allocation at target rate r_i ; idle peer bandwidth allocation; busy peer bandwidth allocation; bandwidth allocation to/from helper pool. Bandwidth allocation in all sub-conferences are coordinated such that bandwidth allocation in any sub-conference advances to stage k only after all sub-conferences finish the allocation in stage $k - 1$.

Stage 1: Video source i allocates r_i bandwidth to send out the video stream it produces. The remaining bandwidth of video source i is updated as $L_i = u_i - r_i$. According to Equation (1), the achieved video rate is $A_i = r_i / |G_i|$.

Stage 2: Video source i utilize idle viewers' bandwidth to grow the achievable video rate from A_i . The detailed algorithm is shown in Algorithm 1. Line 1 picks up an idle viewer p in local sub-conference G_i . Line 2 uses this peer's bandwidth to increase the video rate. $(r_i - A_i)|G_i|$ is the amount of bandwidth needed to improve video rate from A_i to r_i . Line 3 and line 4 update the achieved video rate A_i and the

Algorithm 1 Idle Viewer Bandwidth Allocation

```

1: for each idle peer  $p \in G_i^{(I)}$  do
2:    $x_i = \min((r_i - A_i)|G_i|, L_p)$ 
3:    $A_i = A_i + x_i / |G_i|$ 
4:    $L_p = L_p - x_i$ 
5:   if  $A_i \geq r_i$  then
6:     break
7:   end if
8: end for
9:  $B^{(H)} = B^{(H)} + \sum_{p \in G_i^{(I)}} L_p$ 

```

unallocated bandwidth L_p . Line 5, 6 and 7 break loop if the target rate r_i is achieved. Line 9 allocates the unallocated idle viewer's bandwidth to the helper pool.

Stage 3: In this stage, we allocate the bandwidth on busy peers to sub-conferences in which the target video rate has not been achieved. According to guideline **G3**, a busy peer should first upload to the smaller sub-conference between the one she is viewing and the one she is hosting. To achieve this, we conduct bandwidth allocation for sub-conferences in the no-decreasing order of their sizes. The bandwidth allocation within each sub-conference follows Algorithm 2. This process

Algorithm 2 Busy Viewer Bandwidth Allocation

```

1: for each peer  $p \in G_i^{(S)} \cup \{i\}$  do
2:    $x_i = \min((r_i - A_i)|G_i|, L_p)$ 
3:    $A_i = A_i + x_i / |G_i|$ 
4:    $L_p = L_p - x_i$ 
5:   if  $A_i \geq r_i$  then
6:     break
7:   end if
8: end for
9:  $B^{(H)} = B^{(H)} + \sum_{p \in G_i^{(S)} \cup \{i\}} L_p$ 

```

allocates bandwidth on the video source of sub-conference i and all other viewers who act as video source for other sub-conferences. The allocation is similar to Algorithm 1 and is self-explanatory.

Stage 4: In this stage, a sub-conference that has not achieved its target rate using bandwidth on its source and viewers borrows bandwidth from the helper pool. According to (19) to improve video rate of G_i from A_i to r_i taking into account the helper bandwidth overhead, the needed helper bandwidth is

$$B_i^{(H)} = \frac{(r_i - A_i)|G_i|^2}{|G_i| - 1}.$$

Each sub-conference with bandwidth deficit will request bandwidth $B_i^{(H)}$ from the common helper pool. In the helper pool, if sum of the requested helper bandwidth is not bigger than the aggregate helper bandwidth $B^{(H)}$ contributed by bandwidth surplus sub-conferences, the centralized tracker will allocate to each sub-conference the requested helper bandwidth. Otherwise, the targeted video rate vector is not supportable, and

the tracker can proportionally reduce the helper bandwidth allocation to sub-conferences.

C. Approaching Capacity Region

The algorithm presented in the previous section can be used to check whether a given video rate vector is supportable or not. Through iterative binary search, it can also be used to dynamically approach the max-min capacity γ^* defined in OPT II. We first set the search interval to be $[\gamma_l, \gamma_h]$, with $\gamma_h = \max_{s \in S} u_s$, and γ_l being the lower bounds obtained in Section IV and V. Specifically, for a homogeneous MPVC with normalized upload bandwidth of 1, we set $\gamma_l = \frac{5}{6}$; for a heterogeneous MPVC with normalized average upload bandwidth of 1, we set $\gamma_l = \max(\frac{2}{3}, \frac{|N|}{|N|+|S|})$. From the analysis in Section IV and V, the video rate vector determined by $\gamma_l: \{r_s = \min(u_s, \gamma_l), \forall s \in S\}$, is always achievable. Using γ_l as the starting point, we iteratively find the maximal γ that can be achieved by our bandwidth allocation algorithm. At each iteration, we check whether the video rate vector determined by $\gamma = (\gamma_l + \gamma_h)/2$ is achievable. If yes, the search range shrinks to $[\gamma, \gamma_h]$; otherwise, the search range shrinks to $[\gamma_l, \gamma]$. This process finishes until the range is smaller than a pre-defined threshold ϵ . The binary search pseudocode is presented in Algorithm 3.

Algorithm 3 Approaching Capacity through Binary-search

```

1: procedure MAX- $\gamma(S, N, \{u_i, i \in N\}, \{G_s, s \in S\})$ 
2:   normalize  $u_i$  such that  $\bar{u}_i = 1$ ;
3:   if  $u_i$  homogeneous then
4:      $\gamma_l = \frac{5}{6}$ 
5:   else
6:      $\gamma_l = \max(\frac{2}{3}, \frac{|N|}{|N|+|S|})$ 
7:   end if
8:   while  $(\gamma_h - \gamma_l) > \epsilon$  do
9:      $\gamma \leftarrow (\gamma_h + \gamma_l)/2$ ,  $r_s = \min(u_i, \gamma)$ ,  $\forall s \in S$ 
10:    ok=bandwidth-allocation ( $\{r_s, s \in S\}$ )
11:    if ok==1 then
12:       $\gamma_l \leftarrow \gamma$ 
13:    else
14:       $\gamma_h \leftarrow \gamma$ 
15:    end if
16:  end while
17:  return  $\gamma$ 
18: end procedure

```

VII. NUMERICAL EVALUATION

In this section, we present numerical results to demonstrate the tightness of the derived lower bounds and the efficiency of the proposed bandwidth allocation algorithm.

We adopt three types of performance measures. The first one is the difference between the achieved video rates and the optimal video rates. The second one is the average video quality perceived by all users. Based on the PSNR video

quality model, the average video quality is:

$$\mathcal{V} = \frac{\sum_{i \in N} \log(w_i)}{|N|} = \frac{\sum_{s \in S} |G_s| \log(r_s)}{|N|}, \quad (20)$$

where w_i is the video rate received by viewer i , r_s is the video rate of source s , and $w_i = r_s, \forall i \in G_s$. The third measure is the bandwidth utilization in the conference. First of all, the aggregate received video rate cross all sub-conferences should be less than the sum of upload bandwidth on all peers. Secondly, the video rate of a sub-conference is limited by the bandwidth of its video source. Even if there is abundant bandwidth available, the aggregate received video rate in sub-conference hosted by s is limited by $|G_s|u_s$. We define the upload bandwidth utilization as

$$\mathcal{B} = \frac{\sum_{i \in N} w_i}{\min(\sum_{i \in N} u_i, \sum_{s \in S} |G_s|u_s)} \quad (21)$$

A. Homogeneous One-view MPVC

We first study the tightness of the derived universal lower bounds at different system sizes by varying $|N|$ from 6 to 14 with step-size 4. For each $|N|$, we generate 1,000 random viewing scenarios: we first select a random number of peers as video sources, then each peer randomly selects a source to watch. For each scenario, we first calculate its max-min capacity γ^* using the optimal algorithm OPT II. The CDF distribution of γ^* is plotted in Fig. 4(a). The minimum of γ^* is $0.8333 \approx 5/6$. At all system sizes, more than 90% scenarios have max-min capacity greater than 0.9. Note that the maximum achievable video rate is at most 1. This indicates that while $5/6$ is a universal lower bound independent of viewing relations between peers, for most viewing scenarios, the achievable video rate is pretty close to the upper bound of 1. As the systems size grows, less scenarios can achieve the maximum rate of 1. For each scenario, we also use the binary search algorithm presented in Section VI-C, denoted as the BA algorithm, to iteratively approach the capacity. We also calculate the difference between the achieved rate γ by the BA algorithm with the optimal value γ^* and find the maximum error is smaller than 10^{-3} .

To investigate the impact of the number of sources, we fix $|N|$ at 10 and vary the number of video sources $|S|$ from 2 to 10. For each $|S|$, we generate 1,000 random viewing scenarios and calculate the max-min capacity for each scenario. The results are presented as boxplot in Figure 4(b). For each $|S|$, the central mark in the box is the median, the edges of the box are the 25th and 75th percentiles, the whiskers extend to the most extreme data points not considered outliers, and outliers are plotted individually. When $|S| = 2$, as proved in Theorem 1, the maximal rate of 1 is achieved. As $|S|$ increases, the median value decreases and the variance increases. The lowest median value and the highest variance appear at $|S| = 6$, where the number of possible viewing scenarios is the largest. As $|S|$ increases further, the median increases and the variance decreases. When $|S| = 10$, each peer is a source and only has one reviewer. Each source sends

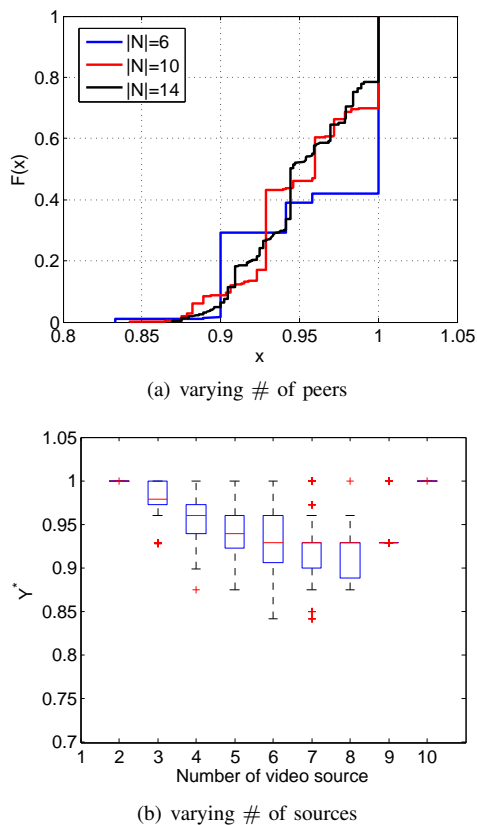


Fig. 4. Capacity of Homogeneous MPVC

TABLE II
BANDWIDTH DISTRIBUTION

	Uplink(kbps)	Probability
class 1	128	0.2
class 2	384	0.4
class 3	1000	0.25
class 4	4000	0.15

video directly to her viewer to achieve the maximum rate of 1.

B. Heterogeneous One-view MPVC

To simulate heterogeneous system, we randomly set peer upload capacity according to the distribution listed in the Table II, which is obtained from a measurement study in [2]. The average peer upload bandwidth is $1.029Mbps$. We vary the number of peers $|N|$ from 6 to 12 with step-size of 2. For each $|N|$, we randomly generate 4,000 viewing scenarios by letting each peer randomly choose another peer to watch. Totally 16,000 random viewing scenarios are generated. Figure 5(a) plots the CDF distribution of max-min capacity obtained by OPT II and BA algorithm. We also plot the lower bound of $\max(\frac{2}{3}, \frac{|N|}{|N|+|S|})\bar{u}_i$ for each scenario. We can see that the BA curve is very close to the OPT II curve. This suggests that the BA algorithm is very efficient in approaching the max-min capacity bound in heterogeneous systems. In the figure, there is a large gap between the max-min capacity and

lower bound. This is because the lower bound is independent of viewing scenarios and is always below peer's average upload bandwidth. But OPT II and BA algorithms work on specific viewing scenario, and the obtained γ reflects the obtained maximal video source rate, which can go well beyond the average upload rate if a video source with high upload bandwidth has just one or few viewers. In Figure 5(a), we also plot the average viewing rate among all peers. In addition to OPT II, BA and the lower bound, we also consider OPT I defined in (6), the bandwidth allocation optimized directly for video quality. The average curves of OPT I, OPT II, and BA algorithm are clustered together, and the gap between them and the average rate curve of the lower bound is smaller than the max-min capacity gap. Figure 5(b) shows the relative performance difference of OPT I, BA algorithm and lower bound compared with OPT II (the relative difference between x and y is define as $\frac{x-y}{y}$). We first consider the max-min capacity obtained by BA. By the curve labeled as " γ^* of BA", the BA algorithm can achieve 93% of optimal max-min capacity with 90% probability. For the average viewing rate, the difference between the BA algorithm and OPT II is fairly small. Since OPT I is optimized for the video quality, the average rate obtained by OPT I can be higher than OPT II. The relative performance of the lower bound is the worst. The average rate of the lower bound is within 75% of OPT II with 80% probability.

Figure 5(c) plots the average video quality \mathcal{V} obtained by different algorithms. The curve of OPT I, OPT II and BA algorithms are almost identical. The performance of the lower bound is worse than the other three algorithms, with the relative difference less than 8%. Finally, Figure 5(d) compares the peer bandwidth utilization \mathcal{B} as defined in (21). The utilization of OPT I, OPT II, BA algorithm are all very close to one. This suggests that those algorithms have efficiently utilize upload bandwidth available on sources and viewers to achieve high video rates, and there is not much space for further quality improvement. But for the lower bound curve, since it is not optimized for specific viewing scenario, the bandwidth utilization is still far from the perfect case. This suggests that the space for bandwidth allocation optimization for individual viewing scenarios is often necessary and rewarding.

To investigate the impact of $|N|$ and $|S|$, we cluster 16,000 random viewing scenarios based on the $\langle |N|, |S| \rangle$ tuple. For each scenario, we normalize the average video viewing rate with the average upload bandwidth. For each $\langle |N|, |S| \rangle$ cluster, we calculate the mean of the normalized average viewing rate for all scenarios in that cluster. Table III presents results for $\langle |N|, |S| \rangle$ clusters with at least 20 random scenarios. Each column corresponds to one system size. Different from the homogeneous case, at all simulated system sizes, the average video rate increases as the number of sources increases. This is because the achieved video rate in each sub-conference is limited by both the source upload bandwidth and the bandwidth available to this sub-conference. When the number of sources is smaller, each source will have more viewers. If a weak peer is chosen as a source, it will degrade the video

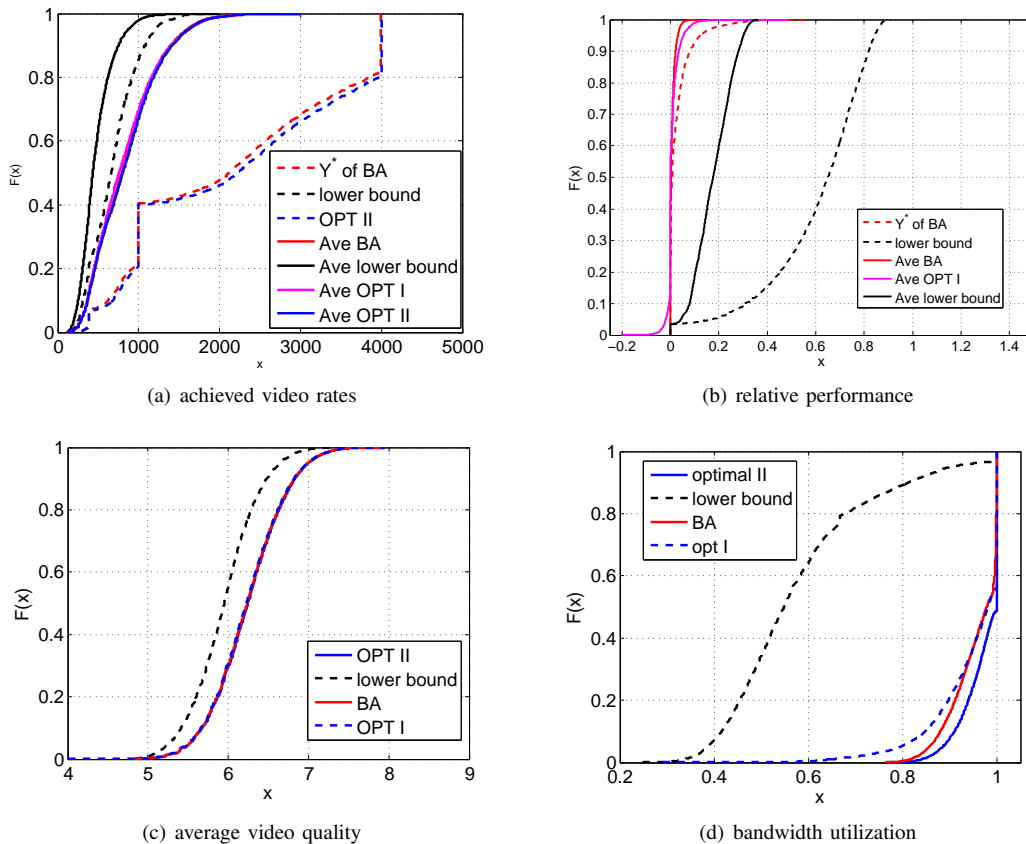


Fig. 5. Performance of Heterogeneous MPVC with 16,000 random viewing scenarios.

TABLE III
HETEROGENEOUS MPVC, RANDOM SOURCES

	$ N =6$	$ N =8$	$ N =10$	$ N =12$
$ S =2$	0.6838	-	-	-
$ S =3$	0.7706	0.7263	-	-
$ S =4$	0.8255	0.7932	0.7700	-
$ S =5$	0.8981	0.8222	0.7753	0.7644
$ S =6$	1.000	0.8610	0.8148	0.8009
$ S =7$	-	0.9088	0.8483	0.8179
$ S =8$	-	-	0.8759	0.8436
$ S =9$	-	-	0.9329	0.8700
$ S =10$	-	-	-	0.8953
$ S =11$	-	-	-	0.9239

quality on more peers. Consequently, the achieved average video rate will be lower.

To eliminate the impact of weak sources, we repeat the previous experiments with an additional requirement that each source must have upload bandwidth larger than the average bandwidth. Specifically, we first generate the peer bandwidth according to Table II, choose only peers with bandwidth larger than the average bandwidth as sources, then let each peer randomly choose a source to watch. According to Corollary 2, we now use $\max(\frac{3}{4}, \frac{|N|}{|N|+|S|})\bar{u}_i$ as the lower bound. The

results are plotted in Fig. 6. When we require all sources have capacity higher than the average upload bandwidth, the source uplink will no longer be the bottleneck. To achieve the max-min fairness, all sub-conferences will achieve the same rate. So the max-min capacity achieved by OPT II is exactly the same as the average viewing rate of all peers. In Fig. 6(a), we only plot the average rates achieved by different algorithms. If we compare Fig. 5(a) and 6(a), we do achieve higher average viewing rates when all sources are bandwidth-rich. But the corresponding max-min capacity γ^* is lower than those achieved in Figure 5(a). This is because when there is no requirement on source bandwidth, sub-conferences hosted by weak sources are limited by source upload bandwidth, strong sources can potentially achieve higher rates and push up the max-min capacity γ^* . Fig. 6(b) plots the relative performance on the average rate of BA, OPT I and lower bound compared with OPT II. Fig. 6(c) compares the average video quality achieved by different algorithms. In Fig. 6(a), 6(b) and 6(c), the new lower bound curves are closer to OPT and BA curves than in Fig. 5(a), 5(b) and 6(c). Comparing Fig. 5(d) and 6(d), bandwidth utilization improves when sources are no longer bottleneck. The lower bound curve in Fig. 6(d) is piecewise constant. This is because the bandwidth utilization defined in (21) is now exactly $\max(\frac{3}{4}, \frac{|N|}{|N|+|S|})$. For the simulated scenarios, there are only limited number of $(|N|, |S|)$ tuples

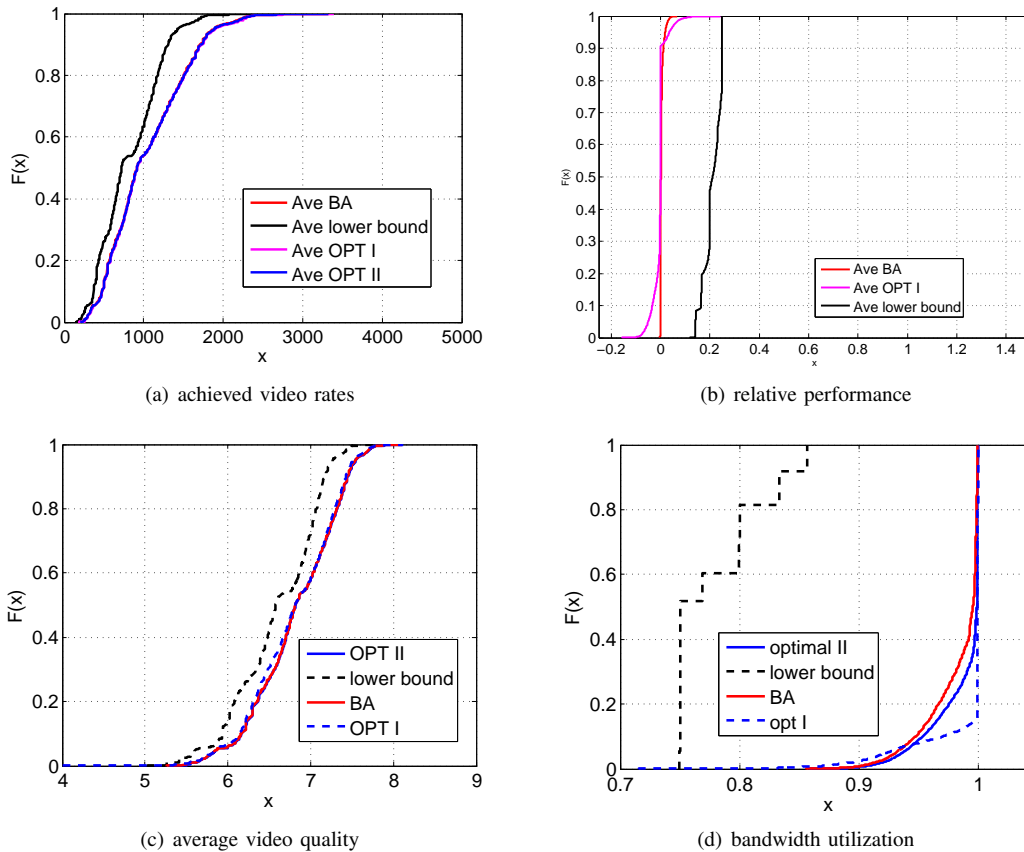


Fig. 6. Performance of Heterogeneous MPVC when each video source's bandwidth is larger than the average bandwidth.

satisfying $\frac{|N|}{|N|+|S|} > \frac{3}{4}$, e.g. $\langle 8, 2 \rangle$, $\langle 10, 3 \rangle$, etc., leading to five discrete values of \mathcal{B} .

Finally, we revisit the impact of the number of sources when sources are bandwidth-rich. As presented in Table IV, opposite to Table III, at all simulated system sizes, when the number of sources increases, the video rate decreases. This is because when the sources are no longer the bottleneck, the achieved video rate in a sub-conference is only determined by the bandwidth available to this sub-conference. When the number of sources is larger, the number of peers in each sub-conference is smaller. With heterogeneous peer upload bandwidth, the average bandwidth within each sub-conference has larger variance. Sub-conferences with less bandwidth have to borrow bandwidth from the helper pool and incur helper bandwidth overhead. Consequently the achieved video rate decreases.

C. Helper Overhead of BA Algorithm

The design objective of the BA algorithm is to achieve target video rates with minimum peer upload bandwidth. The major consideration of the BA design guidelines in Section VI-A is to maximally avoid helper bandwidth overhead. In this section, we study the helper bandwidth overhead incurred by our BA algorithm. We define the aggregate helper bandwidth overhead

TABLE IV
HETEROGENEOUS MPVC, STRONG SOURCES

	$ N =6$	$ N =8$	$ N =10$	$ N =12$
$ S =2$	0.9992	0.999	0.9986	0.9984
$ S =3$	0.9818	0.9791	0.9866	0.9920
$ S =4$	0.9546	0.9702	0.9666	0.9770
$ S =5$	0.9515	0.9500	0.9506	0.9570
$ S =6$	-	-	0.9404	0.9393
$ S =7$	-	-	-	0.9388

ratio as:

$$\mathcal{O}^H \triangleq 1 - \frac{\sum_{i \in N} w_i}{\sum_{i \in S} (u_i^{(s)} + u_i^{(w)} + u_i^{(h)}) + \sum_{i \in I} (u_i^{(w)} + u_i^{(h)})},$$

where in the second term, the numerator is the total video rate received by all peers, and the denominator is the total upload bandwidth consumed on all peers. If there is no bandwidth overhead, the total video receive rate should equals to the total video upload rate.

We generate 100 heterogeneous bandwidth settings randomly according to the bandwidth distribution in Table II. The number of users in the conference is 12. For each each bandwidth setting, we generate 200 random viewing scenarios among peers. For each viewing scenario, we first obtain the

maximal value of γ^* using the BA algorithm. Then we set the target video rate vector as $\{r_s = \min(u_s, \gamma), \forall s \in S\}$, with γ ranging from $0.5\gamma^*$ to γ^* . At each γ , we run our bandwidth allocation algorithm and calculate the incurred helper bandwidth overhead. The distribution of \mathcal{O}^H is shown in Figure 7. In the figure, almost all incurred overhead ratio is less than 25%. This demonstrates that our bandwidth allocation algorithm is robust against random bandwidth settings and viewing relations. The overhead ratio increases as the video rate vector is pushed closer to the capacity bound. This is because to push all sub-conferences to achieve higher video rates, sub-conferences with weak source and viewers have to borrow bandwidth from the helper pool, thus incur higher helper bandwidth overhead.

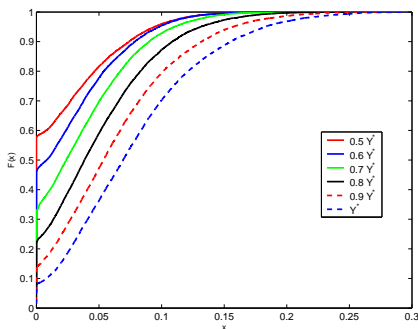


Fig. 7. CDF of Helper Bandwidth Overhead

VIII. CONCLUSION AND FUTURE WORK

In this paper, we explore the design space of pure Peer-to-Peer one-view Multi-party Video Conferencing. We proposed a P2P relay framework for one-view MPVC. Through analysis, we characterized the video rate capacity region of P2P one-view MPVC. For homogeneous system with normalized peer upload bandwidth, we showed that each source is guaranteed to achieve the video rate of $5/6$, that is independent of system size, the number of sources, and the viewing relations between peers. For heterogeneous system with the normalized average peer upload bandwidth, we showed that the guaranteed maximum capacity is $\max(\frac{2}{3}, \frac{|N|}{|N|+|S|})$ with $|N|$ being the number of peers and $|S|$ being the number of sources, which is again independent of the viewing relations between peers. We further showed that all the derived lower bounds are tight. We developed peer bandwidth allocation algorithms that efficiently utilize peers' upload bandwidth to approach the maximal video rate region. Through numerical simulations, we demonstrated that the proposed bandwidth allocation algorithm can achieve a close-to-optimal peer upload bandwidth utilization. Our study demonstrates that P2P is a promising delivery solution for one-view MPVC. Almost all proofs in this paper are constructive and can be applied into real implementation directly with few modifications.

The capacity study here can be generalized to study k -view MPVC where each user watches full videos of k , $1 \leq k \leq$

$|N|$, users. One straightforward way is to decompose a k -view MPVC into k parallel one-view MPVCs, and on each peer, equally partition its upload bandwidth into k shares, one for each one-view MPVC. Then immediately the lower bounds obtained in this paper can be applied to each one-view MPVC after being scaled down by a factor of k . It will be interesting to investigate how much gain one can obtain by considering k -views jointly. Another immediate extension is to study the capacity of server-assisted P2P MPVC, where a server can provide additional bandwidth to disseminate users' videos. To analyze its capacity, we can treat the server as a super peer with abundant bandwidth and randomly assign a source for it to view, then the derived lower bounds automatically apply. Since our derived lower bounds are normalized with the average peer upload bandwidth, the impact of the server assistance is quantified as the increase in the average peer+server upload bandwidth. The lower bounds demonstrate that it is possible to maintain stable video quality on all sources in face of dynamic peer churn and viewing relation changes. We will refine our algorithms to minimize the disruptions to P2P video relays upon peer churn and viewing relation changes.

REFERENCES

- [1] I. E. Akkus, O. Ozkasap, Civanlar, and M. Reha. Multi-objective Optimization For Peer-to-Peer Multipoint Video Conferencing Using Layered Video. In *Packet Video 2007*, 2007.
- [2] A. R. Bharambe, C. Herley, and V. N. Padmanabhan. Analyzing and Improving a BitTorrent Network Performance Mechanisms. In *INFOCOM*, 2006.
- [3] BitTorrent. Homepage. <http://www.bittorrent.com/>.
- [4] M. Chen, M. Ponc, S. Sengupta, J. Li, and P. Chou. Utility Maximization in Peer-to-peer Systems. In *Proceedings of ACM SIGMETRICS*, 2008.
- [5] Y. Chu, S. G. Rao, S. Seshan, and H. Zhang. Enabling Conferencing Applications on the Internet using an Overlay Multicast Architecture. In *Proceedings of ACM SIGCOMM*, 2001.
- [6] M. Civanlar, O. Ozkasap, and T. Celebi. Peer-to-peer multipoint videoconferencing on the Internet. *Signal Processing: Image Communication*, 20(8):4–27, 2005.
- [7] M. Dischinger, A. Haebleren, K. P. Gummedi, and S. Saroiu. Characterizing Residential Broadband Networks. In *Internet Measurement Conference*, 2007.
- [8] Google+. Homepage. <https://plus.google.com/>.
- [9] M. Hosseini and N. D. Georganas. Design of a Multi-sender 3D Videoconferencing Application over an End System Multicast Protocol. In *ACM Multimedia*, 2003.
- [10] Y. Huang, T. Z. Fu, D.-M. Chiu, J. C. Lui, and C. Huang. Challenges, design and analysis of a large-scale p2p-vod system. In *Proceedings of the ACM SIGCOMM*, 2008.
- [11] R. Kumar and K. Ross. Optimal Peer-Assisted File Distribution: Single and Multi-Class Problems. In *IEEE HOTWEB*, 2006.
- [12] J. Lennox and H. Schulzrinne. A Protocol for Reliable Decentralized Conferencing. In *NOSSDAV*, 2003.
- [13] J. Li, P. A. Chou, and C. Zhang. Mutualcast: An Efficient Mechanism for Content Distribution in a P2P Network. In *Sigcomm Asia Workshop*, 2005.
- [14] C. Luo, W. Wang, J. Tang, J. Sun, and J. Li. A Multiparty Video Conferencing System over an Application-Level Multicast Protocol. In *IEEE Transactions on Multimedia*, 2007.
- [15] M. Ponc, S. Sengupta, M. Chen, J. Li, and P. Chou. Multi-rate Peer-to-Peer Video Conferencing: A Distributed Approach Using Scalable Coding. In *Proceedings of ICME*, 2009.
- [16] PPLive. Homepage. <http://www.pplive.com>.
- [17] S. Sen and J. Wang. Analyzing peer-to-peer traffic across large networks. In *IEEE/ACM Transactions on Networking*, pages 219–232, 2002.
- [18] Skype. Homepage. <http://www.skype.com>.

APPENDIX

A. *tightness of lower bound in Theorem 3*

We construct the following homogeneous one-view MPVC: there are six peers with unit bandwidth, four of them are sources, $S = 1, 2, 3, 4$, the viewing relation is: $G_1 = \{3, 4\}$, $G_2 = \{6\}$, $G_3 = \{1, 2\}$, $G_4 = \{5\}$. Plugging in this scenario to OPT II, we obtain the max-min capacity $\gamma^* = 5/6$. Since all sources have bandwidth 1, $5/6$ is the maximal achievable rate on all sources. This proves that the guaranteed capacity C_1 for any homogeneous one-view MPVC can not be higher than $5/6$.