Dealing with User Heterogeneity in P2P Multiparty Video Conferencing: Layered Coding Versus Receiver Partitioning

Eymen Kurdoglu, Yong Liu and Yao Wang
Department of ECE, NYU Polytechnic School of Engineering

***Abstract*—Layered coding elegantly handles user bandwidth heterogeneity in video conferencing, however, it incurs rate and complexity overheads. An alternative is partitioning the receiver sets and using non-layered coding for each group. In this paper, we investigate how to maximize the received video quality for both systems under uplink and downlink capacity constraints, while limiting the number of hops that the videos travel by two. Towards this end, we first show that any multicast tree is equivalent to a collection of depth-1 and depth-2 trees, under outbound *and* inbound flow constraints. For the layered system, we propose an algorithm that simultaneously solves for the number of video layers, the rate and distribution tree of each layer. For the receiver partitioning system, we develop an algorithm for determining the receiver partitions and tree construction for each group. Through numerical comparison study, we show that the receiver partitioning system achieves significantly higher video quality than the layered system, due to its higher coding efficiency.**

I. Introduction

Advances in broadband and video encoding technologies have enabled Multi-Party Video Conferencing (MPVC) applications [1], [2], [3] to flourish on the Internet. Most existing MPVC solutions are *server-centric* [4], ignoring the network and geographic locality of users in the same conference. Users located far away from servers traverse long-delay paths, leading to poor conferencing experience. The natural delivery solution for MPVC is *Peer-to-Peer (P2P)*, where users send their data to each other directly. P2P MPVC solutions have recently been proposed [5], [6], [7], [8], [9] to offload servers and exploit user locality to achieve low delay and high rate.

P2P MPVC, where multiple users multicast voice and video with intense bandwidth requirements in real-time, has to deal with the inherent heterogeneity of users in the same conference, in terms of upload/download capacity, computation and energy supply. To deal with peer heterogeneity, it is very important to design video generation and distribution in an *integrated* fashion. One solution is *layered coding*, where each source encodes multiple video layers using the recent layered video coding techniques [10], and receivers downloading more layers will receive better video quality. An alternative solution is *receiver partitioning*, where each source generates multiple video versions using the traditional single-layer video coding techniques, receivers are partitioned into different groups, with receivers in each group receive the same video version. For P2P MPVC, layered coding and receiver partitioning enable different P2P sharing opportunities. With layered coding, receivers receiving different subsets of layers can still share their common video layers, leading to a higher P2P sharing efficiency. With receiver partitioning, only receivers watching the same version can share video with each others. However, the flexibility of layered coding comes at the price of non-negligible rate overhead, that is, to achieve the same perceptual quality, layered coding has to use higher bit rate than non-layered coding [10]. Layered coding also has much higher encoding and decoding complexity, and consumes more CPU cycles and energy, which limits its adoption by mobile devices.

In this paper, we study the achievable performance by layered coding and receiver partitioning in P2P MPVC. Instead of assuming a layered coding scheme with zero overhead like most P2P MPVC studies, we consider realistic layered coding schemes with practical encoding overhead ratios obtained using the H.264/SVC codec [10]. Moreover, almost all existing P2P MPVC studies [5]-[9] assume that peer downlinks are *never* bottlenecks. Such an assumption is too crude to model user heterogeneity in MPVC, in which each user needs to download multiple streams, so the downlink may potentially become a bottleneck for some users, even if their downlink capacity is higher than their uplink capacity. This scenario becomes more severe with wireless/mobile users.

Our main contributions are summarized as the following: (1) We first develop the general formulation for tree-based P2P MPVC distribution *under both peer uplink and downlink capacity constraints*. We show that any distribution tree can be reduced to a collection of depth-1 and depth-2 trees, which greatly reduces the computational complexity for searching the optimal trees. (2) For layered coding, we design an integrated video encoding and distribution algorithm that simultaneously solves for the number of video layers and the rate of each layer to be generated on each source, as well as the subset of layers each receiver should receive from each source. (3) For receiver partitioning, we study the optimal receiver partitioning problem which partitions receivers of the same source to multiple groups and uses single-layer coding in each group. We propose a fast heuristic algorithm to solve it. (4) We compare the performance of layered coding and receiver partitioning through numerical simulations. In our simulations, layered coding can always achieve the optimal video rates. And somewhat surprisingly, our receiver partitioning heuristic can also achieve close-to-optimal video rates in most simulated cases. In terms of video quality, due to the substantial rate overhead of layered coding, receiver partitioning significantly outperforms the layered coding.

II. Problem Statement and Multicast Trees

A. General Formulation for P2P MPVC

We examine an MPVC scenario where each participant transmits its own video to all other participants. The end-users are connected through an overlay P2P network , which is represented by a node-capacitated, complete, directed graph $G = (N, A)$, and $|N| = n$. Henceforth, we use the terms users, nodes, peers and participants interchangeably. Our assumption that each node maintains $n-1$ connections with the rest of the nodes can be justified due to the small number n of participants in a typical MPVC session. Only the user uplink and downlink capacities create bottlenecks in the network, bounding the incoming and outgoing flows of a node. We label the users

$\{1, \ldots, n\}$ in ascending order of their download capacities so that node $i \in N$ has upload and download capacities U_i and D_i, respectively and the downlink capacities are ordered as $D_1 \leq \cdots \leq D_n$. Each user concurrently hosts an application layer multicast session to distribute its own video sequence to others, i.e., the receiver set R_i for node i is $N \setminus \{i\}$.

As for the video encoding methods, we consider layered and non-layered techniques. Non-layered encoding of a video sequence produces a single video stream, where the video chunks must be received entirely before decoding. Thus, all the users in a non-layered video multicast group receive the same video stream at the same bit rate. Unlike non-layered coding, layered video encoding [10] produces a bitstream that consists of multiple layers that can be decoded progressively. Video chunks consist of sub-chunks that correspond to different quality layers that can be decoded in a nested fashion, starting from the base layer and then the enhancement layers. Therefore, the users may receive the same video sequence at different rates. The cost of this flexibility is what is called the coding overhead of layered video, which is defined as the additional video rate needed to achieve the same quality as a non-layered coder. For example, to achieve the same video quality, the H.264/SVC coder, which is a layered coding standard, requires up to 30% higher video rate than the H.264/AVC coder, which is a single-layer coding standard (Fig. 2).

In order to multicast its video, peer i makes use of a number of directed distribution trees, which we denote generally by $T \in \mathcal{T}$. These multicast trees are rooted at peer i itself and spanning, in general, a subset of the receiver set R_i. Packets originating from the sources are routed along these trees, where nodes on the trees replicate the packets and send them to their downstream nodes. We denote the source node of a directed tree T by $s(T)$ and its vertex set by $V(T)$. We assume that the packet flow rate, denoted by $x(T)$, is equal along all arcs of a given tree T, since any tree with unequal flows on its arcs can be decomposed into sub-trees with equal flows. Thus, the total communication rate r_{ij} from user i to user j is simply the sum of the rates of trees that are rooted at node i and cover node j,

$$r_{ij} = \sum_{T:\ s(T)=i,\ j \in V(T)} x(T). \tag{1}$$

Therefore, for an arbitrary set $\mathcal{T}$ of multicast trees and an arbitrary concave utility function f that measures the video quality at rate r, a general application layer flow configuration problem can be formulated as

$$\max_{\substack{x(T) \geq 0 \\ \forall T \in \mathcal{T}}} \sum_{i \in N} \sum_{j \in R_i} f\left(r_{ij}\right) \quad \text{subject to} \tag{2}$$

$$\sum\nolimits_{\forall T \in \mathcal{T}:\ i \in V(T)} c(i,T) x(T) \leq U_i, \quad \forall i \in N \tag{3a}$$

$$\sum\nolimits_{j:\ i \in R_j} r_{ji} \leq D_i, \quad \forall i \in N, \tag{3b}$$

where r_{ij} is the rate of node i's video that node j receives. $c(i,T)$ is the number of children nodes that node i has on tree T and r_{ij} is given as in Eq(1). The notation used is summarized in Table I. For simplicity, we assume that the videos from all participants in a conference have similar characteristics and thus simply use $f(r)$ to represent the quality-rate relation of such a video. The problem with this formulation is that the number of potential trees that can be considered is very large. Note that the following constraints always hold for any multicast tree set chosen in any P2P MPVC system.

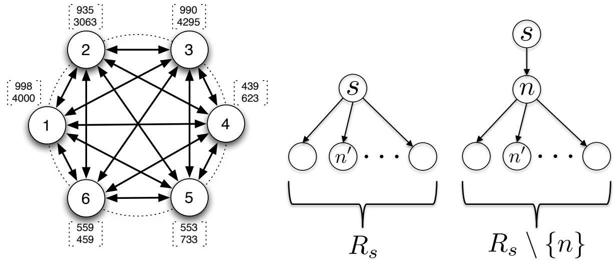

Fig. 1. An example MPVC overlay, with given upload & download capacities (left), depth-1 and depth-2 Mutualcast (MC) Trees (right)

$$\sum\nolimits_{(i,j)} r_{ij} \leq \sum\nolimits_{m} U_m \tag{4a}$$

$$\sum\nolimits_{i} r_{ij} \leq D_j \text{ and } \max_j(r_{ij}) \leq U_i, \quad \forall j \in N \tag{4b}$$

Therefore, the region defined by these constraints presents, in general, a loose upper bound on the achievable video rates.

B. Optimal Multicast Trees

A crucial design problem for a P2P-based MPVC system is then to determine which trees should be used in a given node-capacitated complete graph. It was shown in [6] that employing the two types of trees in Figure 1, introduced in [5], is sufficient to maximize the throughput and the utility in a multi-source P2P scenario without helper nodes, under the assumption that the network is *uplink-throttled*. Hereafter, we call such trees *Mutualcast (MC) trees*. MC trees for node i are all rooted at i and consists of a 1-hop tree that reaches all $j \in R_i$ and R_i 2-hop trees, each passing through a particular $j \in R_i$ and then branching to the rest. In such an uplink-throttled setting, all receivers of a source node s receive the video at the same rate. However, to the best of our knowledge, there is no optimality result for any trees in an uplink- *and* downlink-throttled network. At this point, we present the following theorem, which shows that any given tree with flow f can be replaced by MC trees covering the same node set as before.

Theorem 1. *A node-capacitated, directed multicast tree $T = (N \cup \{s\}, E)$ rooted at s can be replaced by 1-hop and 2-hop MC trees that are rooted at s and span $N \cup \{s\}$, and the aggregate download and upload rate of each node in all the MC trees are exactly the same as in the original tree T.*

Proof: Refer to [11]. ■

Remark 1. *In a node-capacitated, directed, complete graph $G = (N, A)$, any feasible flow configuration achieved by a given set of trees, each of which spans a subset of N, can also be achieved by a combination of 1-hop and 2-hop MC trees that cover the same subsets. Thus, to find the optimal set of multicast trees, it is sufficient to consider only MC trees.*

III. Design of Layered System

In this section, we look into the problem of layered video distribution in a fully-connected P2P network with upload and download constraints. We first describe the multicast tree sets to be employed. Afterwards, we formulate the optimal flow configuration problem as a tree-packing problem with

N	Set of participants in the conference
U_i	Upload capacity of peer i
D_i	Download capacity of peer i
R_i	Receiver set for peer i
$x(T)$	Rate of tree T
r_{ij}	Total video rate from peer i to peer j
T_{iik}	1-hop Mutualcast tree rooted at i, delivering layer k
T_{ijk}	2-hop Mutualcast tree rooted at i, going through j and delivering layer k
x_{ijk}	$x(T_{ijk})$
$S_l^{(i)}$	Peers receiving l^{th} layer of peer i's video
L_i	Number of layers in peer i's video
b_{ik}	Part of U_i allocated to drive layer k distribution trees
z_{ik}	Bitrate of layer k of peer i's video
P_i	A given partition of R_i
$G_k^{(i)}$	k^{th} group in partition P_i
$g_k^{(i)}$	Bitrate of peer i's video received by users in $G_k^{(i)}$
$b_k^{(i)}$	Part of U_i allocated to receivers in $G_k^{(i)}$
u_{ij}	Part of U_i allocated to help disseminate user j's video

TABLE I
NOTATION REGARDING FORMULATION, LAYERED VIDEO SYSTEM AND RECEIVER PARTITIONING SYSTEM

continuous rates. For this study, we assume that a video can be coded into an arbitrary number of layers and that each layer can have any rate that varies over a continuous range. We recognize that this may not be feasible in practice, but analysis based on this idealistic assumption obtains performance upper bound for layered coding and provides important insight for our comparison study.

A. Determination of Subscriber Sets

According to Remark 1 above, we only need to determine the sets of nodes to be spanned by the MC trees to find an optimal flow configuration. Since all nodes in a tree share a common packet flow, determining which nodes to span depends on the video stream structure. Specifically, for a layered video stream, these sets of nodes are, in fact, receiver sets of particular video layers. Denote the set of users that receive the l^{th} layer of node i's video by $S_l^{(i)}$, usually referred to as *subscribers* of the l^{th} layer in the literature. Then, only the l^{th} video layer is distributed through the multicast trees that span $S_l^{(i)}$. Due to layered coding, $S_l^{(i)}$ have to be nested, since the nodes need to receive all the layers up to $l-1$ in order to decode the l^{th} layer. As a result, all receivers subscribe to the first (base) layer. Thus, for user i's video stream $\mathbb{V}_i$, we have $S_{L_i}^{(i)} \subset S_{L_i-1}^{(i)} \subset \cdots \subset S_1^{(i)} = R_i$, where L_i is the number of video layers that user i generates. We do not assume that L_i is given, nor that it is bounded by source capabilities or user preferences. Rather, L_i is only bounded by the number of receivers R_i and will be determined with the following subscriber determination heuristic. For $S_l^{(i)}$, nodes in R_i are sorted in ascending order of their total download capacities. Clearly, $S_1^{(i)} = R_i$. Next, we remove the node(s) with the smallest total download capacity. The remaining nodes make up $S_2^{(i)}$, i.e., they are receivers of layer 2. We proceed in this fashion until every node is removed. With this heuristic, the number of layers for source i equals to the number of receivers for source i, if all receivers have different downlink capacities. Once the subscriber sets and the number of video layers are determined for source peer i, each layer l of $\mathbb{V}_i$ is distributed with the help of $|S_l^{(i)}|$ 2-hop trees $\{T_{iml} : m \in S_l^{(i)}\}$ and the single 1-hop tree T_{iil}. All these trees allow the users to share their upload bandwidth with other users, increasing throughput and network utility. Let us define z_{il} as the rate of layer l of user i's video $\mathbb{V}_i$. Then,

$$z_{il} = x_{iil} + \sum\nolimits_{j\in S_l^{(i)}} x_{ijl} \tag{5}$$

Finally, if b_{il} is the upload bandwidth that user i requires to drive its own layer l distribution trees into $S_l^{(i)}$, we have

$$\begin{aligned} b_{il} &= |S_l^{(i)}| x_{iil} + \sum\nolimits_{j\in S_l^{(i)}} x_{ijl} && (6) \\ &= \left(|S_l^{(i)}| - 1\right) x_{iil} + z_{il}. && (7) \end{aligned}$$

B. Problem Formulation

In this section, we are finally ready to formulate the multi-source, multi-rate flow optimization problem for layered videos, given the layer subscriber sets for each video source.

$$\max_{x_{ijk}\geq 0} \quad \sum\nolimits_{i\in N} \sum\nolimits_{j\in R_i} Q_{LV}\left(r_{ij}\right) \quad \text{subject to} \tag{8}$$

$$\sum_{k=1}^{L_i} b_{ik} + \sum\nolimits_{j\neq i} \sum\nolimits_{k:\ i\in S_k^{(j)}} \left(|S_k^{(j)}| - 1\right) x_{jik} \leq U_i \tag{9a}$$

$$\sum\nolimits_{j\neq i} r_{ji} \leq D_i, \quad \forall i \in N \tag{9b}$$

$$r_{ij} = \sum\nolimits_{k:\ j\in S_k^{(i)}} z_{ik}, \quad \forall (i,j), i \neq j \tag{9c}$$

(9a) follows since a video source can allocate part of its upload bandwidth to relay its own video layers and part of it for helping the other sources for which itself is a receiver. The objective function Q_{LV} given in (8) is a non-decreasing, concave function of r_{ij}. Furthermore, feasible region defined by inequalities (9a)-(9c), (5) and (7) is a convex polytope in tree variables. Using Eq. (1), we can introduce the video rate variables r_{ij} in the inequalities and then take the projection of the polytope onto the $\{r_{ij}\}$ coordinates. Projection preserves convexity, therefore the achievable rate region is also convex. As a result, the optimization problem in (8)-(9) is a non-strictly concave optimization problem in the tree rate variables and has an infinite number of solutions. However, if there exists an interval I such that Q_{LV} is strictly concave in I and the optimal video rates lie in I, then they are unique, hence the layer rates z_{ik} are also unique. Centralized solution techniques for such concave optimization problems have been well-understood and hence, any one of these solution methods can be employed to find a solution.

In this formulation, the number of variables (number of trees employed) is $O(n^3)$ in the worst case. However, employing a large number of multicast trees could lead to increased jitter in a practical implementation. Therefore, after finding the optimal vector of video rates $\boldsymbol{r}^*$ that maximizes the network-wide video quality, it is of interest to find a configuration of tree rates x_{ijk} that achieves $\boldsymbol{r}^*$ and favors 1-hop trees instead of 2-hop trees, as the packets that are distributed through 1-hop trees suffer less end-to-end delay. In [11], we present a method to find a set of feasible tree rates that satisfies the constraints and achieves a given vector of video rates r_{ij}, while favoring 1-hop trees over 2-hop trees.

It is known that layered video encoding methods present a higher computational complexity than their non-layered counterparts, which might be limiting for mobile devices with computation and power constraints. Furthermore, the use of layered coding has the disadvantage of the *coding overhead*, which will be discussed in the next section. Therefore, we now turn our attention to MPVC systems where the users employ non-layered video coding techniques, with the hopes of finding an alternative that overcomes these problems.

IV. Design of Receiver Partitioning System

Although layered coding enables generating a flexible stream that offers variable qualities depending on the rate, the cost of such flexibility is an increased bit rate to achieve a certain quality, which is referred to as the *coding overhead* of layered encoding. As an example, Fig. 2 presents the normalized subjective quality vs. bit rate curves of the Crew video sequence obtained by using H.264/AVC (non-layered) and H.264/SVC (layered) standards, respectively. Coding overhead of layered encoding (up to 30% at some rates), along with its relatively higher computational complexity, motivates the use of non-layered video in MPVC systems.

Clearly, multicasting the same non-layered video to all receivers is suboptimal, starving the receivers with higher download capacities. In order to obtain a multi-rate solution, a source can generate multiple video versions and send different video versions to different users at different rates, matching their download capacities. The drawback of this method in terms of bandwidth is that the source may not have sufficient upload capacity to send out different streams in the first place. Accordingly, we propose creating receiver partitions in each of the n multicast sessions, where the nodes in each group within a partition can share their upload bandwidth using MC trees rooted at the source of the session and spanning all the nodes in the group.

A. Problem Formulation

Now, let R_i be partitioned such that the groups in the partition are denoted by $G_k^{(i)}$ and $P_i = \{G_k^{(i)}, k = 1, \ldots, K_i\}$ is the partition with $\bigcup_k G_k^{(i)} = R_i$. Each node j in a given group $G_k^{(i)}$ receives the video at the same *group rate* $g_k^{(i)} = r_{ij}$, but the nodes in different groups have different rates. Hence, the users with higher download capacities can receive more, resulting in a higher average video quality. Now, assuming that we are given a specific collection $P = \{P_i : i \in N\} \in \mathcal{P}$, where P_i is the partition of receivers of source i, we can formulate the multi-source, multi-rate video quality maximization problem with non-layered encoding as,

$$\max_{u_{ij}, b_k^{(i)}, g_k^{(i)} \geq 0} \quad \sum_{i \in N} \sum_{k=1}^{K_i} |G_k^{(i)}| Q_{NL}(g_k^{(i)}) \quad \text{subject to} \tag{10}$$

$$g_k^{(i)} \leq b_k^{(i)}, \quad \forall i \in N, k = 1, \ldots, K_i \tag{11a}$$

$$|G_k^{(i)}| g_k^{(i)} \leq b_k^{(i)} + \sum\nolimits_{j \in G_k^{(i)}} u_{ji}, \quad \forall i \in N, \forall k \tag{11b}$$

$$\sum\nolimits_{k=1}^{K_i} b_k^{(i)} \leq u_{ii}, \quad \forall i \in N, k = 1, \ldots, K_i \tag{11c}$$

$$\sum\nolimits_{j=1}^{n} u_{ij} \leq U_i, \quad \forall i \in N \tag{11d}$$

$$\sum\nolimits_{j:\ i \in G_k^{(j)}} g_k^{(j)} \leq D_i, \quad \forall i \in N. \tag{11e}$$

Here, $b_k^{(i)}$ and u_{ij} denote the portions of the total upload capacity of node i that is allocated for use in its own multicast group $G_k^{(i)}$ and in $G_k^{(j)}$, where $i \in G_k^{(j)}$, respectively. Again, the objective function Q_{NL} in (10) is a non-decreasing, concave function of the video rate and the feasible region defined by (11) is convex. Hence, the formulated problem above is a non-strictly concave optimization problem with linear constraints. Similar to (8), it has a unique solution in the group rates $g_k^{(i)*}$, assuming the optimal solution lies where Q_{NL} is strictly concave, whereas u_{ij}^* and $b_k^{(i)*}$ are not unique. The number of variables, which depends on P, is $O(n^2)$ in the worst case. Determination of the MC tree rates is straightforward once we have u_{ij}^* and $b_k^{(i)*}$; the 2-hop multicast tree rooted at node i and passing through node $j \in G_k^{(i)}$ has rate $x_{ij} = u_{ji}/(|G_k^{(i)}| - 1)$ and for the 1-hop tree we have $x_{ii} = (b_k^{(i)} - \sum_{j \in G_k^{(i)}} x_{ij})/|G_k^{(i)}|$.

The difficulty with employing non-layered coding in MPVC systems is that we do not readily know the optimal P^*. The size $|\mathcal{P}|$ of the set of all receiver partition collections is given by $(B_{n-1})^n$, where n is the number of participants and B_m is the m^{th} Bell number, equal to the number of ways a set of cardinality m can be partitioned. Therefore, exhaustively searching among all possible collections of partitions is hopeless even for a small number of users. In order to overcome this difficulty, we now propose a simple heuristic algorithm to find a suitable collection of receiver partitions, as well as the group rates that can be achieved.

B. Heuristic Algorithm for Partitioning

The main idea behind the heuristic is to shrink the search space by decomposing the problem of finding the best collection P^* of partitions into separate problems of finding the best partition P_i^* for each source $i \in N$. We start our analysis by assuming that a set of target video rates $\{r_{ij} : \forall (i,j), i \neq j\}$ is given. Let us define the total rate needed to multicast source i's video as $M_i = \sum_{j \in R_i} r_{ij}$. Note that the benefit of using 2-hop multicast trees is that a peer i can still sustain a video session with total multicast rate M_i, greater than its own upload bandwidth U_i, by exploiting the other peers with abundant upload bandwidths. In this case, additional bandwidth required to drive peer i's video session would simply be $S_i = M_i - U_i > 0$, which can also be thought of as the net bandwidth shift into user i's video session. Let us define the set $\boldsymbol{\epsilon} = \{i \in N : S_i > 0\}$ and call such peers *ϵ-peers*. If we have an ϵ-peer, there must be another peer j with $S_j < 0$, otherwise we would have $\sum_{m \in N} U_m < \sum_{(i,j)} r_{ij}$. Let us then call such peers, which provide additional bandwidth, *α-peers*; $\boldsymbol{\alpha} = \{i \in N : S_i < 0\}$. It is sufficient for each α-peer to unicast its video to each of its receivers at rate r_{ij}.

As hinted above, determination of the α- and ϵ-peers, as well as the bandwidth shifts between them, is critical in order to find a good solution. Let s_{ij} denote the amount of bandwidth provided by node i to node j. In our heuristic algorithm, we only allow bandwidth shifts to occur from α-peers to ϵ-peers, and the optimal values of these are estimated through the solution of the following optimization problem.

$$\max\nolimits_{r_{ij} \geq 0} \quad \sum\nolimits_{(i,j)} Q_{NL}(r_{ij}) \tag{12}$$

subject to the constraints given in (4).

It is necessary for *any* feasible $\{r_{ij}\}$ to satisfy the constraints in (4). Therefore, the optimal solution of (12) under constraints defined in (4) gives also a quality upper bound for any achievable video rate under the given upload and download constraints. Note that this problem can be easily solved using a simple water-filling algorithm: each source sends out equal flows to each receiver, while gradually increasing the flows at the same pace, until either all peers are downlink-saturated or there is no more upload bandwidth. We then calculate $S_i = M_i - U_i$ for each $i \in N$ and classify the peers accordingly. Each α-peer i offers in total $|S_i^*|$ units of bandwidth, whereas each ϵ-peer j requires S_j^* additional units of bandwidth to support its total multicast rate M_j. The algorithm distributes the bandwidth provided by the α-peer i to ϵ-peer j proportionally. So, we have

$$s_{ij} = \begin{cases} S_j \frac{S_i}{\sum_{m \in \boldsymbol{\alpha}} S_m}, & \text{if } i \in \boldsymbol{\alpha} \text{ and } j \in \boldsymbol{\epsilon} \\ 0 & \text{otherwise.} \end{cases} \tag{13}$$

Next, we limit the total rate that a peer is allowed to receive in a particular video session by dividing the download capacity of each peer equally between the video sessions. Thus, for source i, a receiving peer j has a download capacity of $D_j/(n-1)$. All that remains is finding a suitable receiver partition for each source. For any source i, this is performed by searching only through the ordered partitions, starting with the single-group partition $P_i = \{R_i\}$ that includes all the receivers. Here, a receiver partition $P_i = \{G_k^{(i)}, k = 1, \ldots, K_i\}$ is ordered if we have $D_k \leq D_{k'}$ for all $k \in G_\ell^{(i)}$, $k' \in G_{\ell'}^{(i)}$ and $\ell < \ell'$. The remainder of our heuristic can be regarded as a distributed algorithm, as each user i performs a search to find a suitable receiver partition on R_i. A steepest-ascent hill climbing method is employed by each user i to search for a local maximum by examining the neighboring ordered partitions. We consider two partitions as neighbors if only if the number of groups they contain differs by at most one. At each step of the greedy search, only the neighbor partitions containing one more group are examined. More specifically, for each candidate partition P_i, peer i solves the following optimization problem.

$$\max_{g_k^i \geq 0} \quad \mathbb{Q}(P_i) = \sum\nolimits_{k=1}^{K_i} |G_k^{(i)}| Q_{NL}(g_k^{(i)}) \quad \text{subject to} \tag{14}$$

$$g_k^{(i)} \leq b_k^{(i)}, \quad k = \{1, \ldots, K_i\} \tag{15a}$$

$$|G_k^{(i)}| g_k^{(i)} \leq \begin{cases} b_k^{(i)} + \sum_{j \in G_k^{(i)}} s'_{ji} & \text{if } |G_k^{(i)}| > 1 \\ b_k^{(i)} & \text{if } |G_k^{(i)}| = 1 \end{cases} \tag{15b}$$

$$\sum\nolimits_{k=1}^{K_i} b_k^{(i)} \leq U_i^{(\text{eff})} \tag{15c}$$

$$s'_{ji} \leq s_{ji}, \quad \forall j \in R_i \tag{15d}$$

Here, $U_i^{(\text{eff})}$ is the effective upload capacity that node i is allowed to use. Clearly, if node i is an α-peer, $U_i^{(\text{eff})} = M_i$, otherwise we have $U_i^{(\text{eff})} = U_i$. As before, $b_k^{(i)}$ is the portion of the effective upload capacity of user i that is allocated to group $G_k^{(i)}$ and s'_{ji} is the bandwidth provided to node i by node j, bounded by s_{ji}. If there is no α-peer in the group, then s_{ji} is necessarily zero, since we only allow the α-peers to provide bandwidth for the video distribution. After the examination of all the ordered neighboring partitions, the one that yields the highest average session quality $\mathbb{Q}(P_i)$ is selected as the new local maximum candidate. The algorithm stops when there is no neighbor partition that yields a higher average session quality. Note that the distributed phase of the whole process, which is summarized in the pseudocode of Algorithm 1, can be easily implemented in any device by using any convex optimization algorithm.

Algorithm 1 Receiver partition selection heuristic

```
Find the solution of (12)-(4)                              ▷ Initialization
Calculate $S_i = M_i - U_i$ for $\forall i \in N$
Create matrix $[s_{ij}]$ according to Eq.(13)
for all $i \in N$ do                                       ▷ Distributed phase
    $P_i^{(best)} \leftarrow \{R_i\}$, $\mathbb{Q}_i^{(best)} \leftarrow \mathbb{Q}(\{R_i\})$
    repeat
        $P_i^{(current)} \leftarrow P_i^{(best)}$, $\mathbb{Q}_i^{(current)} \leftarrow \mathbb{Q}_i^{(best)}$
        for all $P_i \in$ neighbors($P_i^{(current)}$) do
            Find the solution of (14)-(15)
            if $\mathbb{Q}(P_i) > \mathbb{Q}(P_i^{(current)})$ then
                $P_i^{(best)} \leftarrow P_i$, $\mathbb{Q}_i^{(best)} \leftarrow \mathbb{Q}(P_i)$
            end if
        end for
    until $P_i^{(current)} = P_i^{(best)}$
end for
```

V. Simulation Results

In this section, we numerically evaluate the capacity regions achievable through layered coding and receiver partitioning. It is worth restating that our focus is completely on maximizing the video quality of P2P MPVC under two different design choices and we do not explicitly minimize delay. However, end-to-end packet delay in our overlay is controlled since a packet goes through at most 2 hops. In our simulations, we consider layered video distribution scheme, given as the solution of (8)-(9) in Section III, receiver partitioning video distribution scheme, given as the solution of (10)-(11) with optimal partitions in Section IV and the fast heuristic algorithm presented in Algorithm 1 in Section IV-B, along with the straw-man schemes of multiple unicast and single-rate multicast. For the clarity of comparison, we will assume that the ratio of the download capacity of a peer to its upload capacity is the same for all peers in the video conference, that is, $w = D_i/U_i$, for all $i \in N$. In all our simulations, we assume the network is static within the time needed to perform the rate optimization. Participants encode their videos according to H.264/AVC (non-layered encoding) and H.264/SVC (layered encoding) standards. In video conferences, users' video sequences are likely to have similar features, therefore we associate the same video quality-rate function with each user. Specifically, in our simulations, we use the following normalized subjective quality model presented in [12]; $Q(r) = \frac{1-e^{-\kappa(\frac{r}{r_{max}})}}{1-e^{-\kappa}}$, where r is the received video rate, and κ and r_{max} are parameters that depend on the video characteristics and layer configuration. r_{max} is the video rate needed to code the video at the highest quality (achieved at the highest spatial, temporal, and amplitude resolutions considered). As an example, subjective quality of the Crew video sequence with respect to the bit rate for both H.264/AVC and H.264/SVC encodings can be seen in Figure 2. The sequence is encoded at 5 temporal, 4 quantization and 3 spatial resolutions. For a given rate, the optimal spatial, temporal and amplitude resolutions that maximize the perceptual quality are chosen. For this example, the quality-rate model has the following parameters: $\kappa_{\text{SVC}} = 3.121$, $\kappa_{\text{AVC}} = 3.4$, $r_{max}^{\text{AVC}} = 2969$ kbps and $r_{max}^{\text{SVC}} = 3515$ kbps.

We examine a video conferencing scenario where the par-

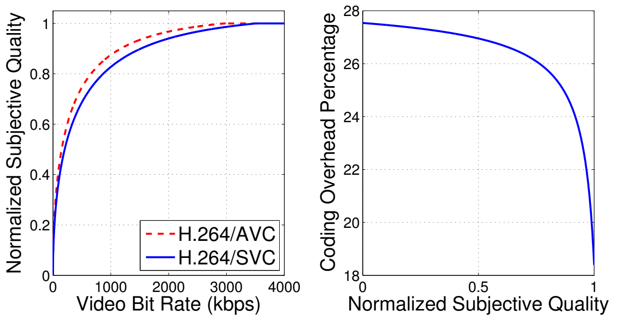

Fig. 2. Quality vs. rate curves of the Crew sequence encoded according to H.264/AVC and H.264/SVC standards (left), additional bit rate (percentage) required by H.264/SVC to achieve the same quality as H.264/AVC (right).

ticipants are chosen from 4 different user classes with respect to their upload capacities, considered as 500, 3500, 6500 and 9500 Kbps for each class. Note that these capacities are selected to reflect the performance of cellular data, wifi and wired users. The download capacities can then be calculated for different w. We randomly pick 6 users out of these classes with a uniform distribution and generate the average quality curves for $0.1 \leq w \leq 5$ by averaging over 50 randomly selected bandwidth profiles (Fig. 3). Our findings with more users are virtually the same. Due to its complexity, we exclude the optimal partitioning scheme from each of these simulations. On the left hand side, the average quality performances are shown. On the right, the average quality curves normalized with respect to the performance of the maximum bound solution obtained in (12) are shown, along with the average total throughput. It is seen that SVC dissemination achieves a lower average video quality, due to the coding overhead. In order to show the effect of the coding overhead on the video qualities, we also show the performance of layered video distribution without any coding overhead, which, in all cases, results in the best quality, although with a small difference. Furthermore, we observe that the optimal receiver partitioning strategy with non-layered video distribution is almost as good as layered video distribution without coding overhead. *This result shows that, in MPVC systems where the downlinks and uplinks may both present bottlenecks, we can obtain a multi-rate solution by using optimal receiver partitioning and non-layered video without any significant performance loss in terms of the average or minimum video quality, compared even with an ideal layered video distribution scheme with no overhead*. Among all solutions, the proposed partition heuristic comes the closest to the maximum bound in terms of the achieved video quality, although its average video rate falls slightly below the theoretical bound for the average rate, as seen at bottom right in Figure 3. This rate degradation happens around $w = 1$, where the total download capacity is equal to the total upload capacity. Moreover, although the average rates for all schemes are close to the theoretical maximum bound (except for the single-rate method), the delivered video qualities differ significantly. Especially for layered coding, the achieved rate is as high as the bound, but again the achieved quality is discounted by the coding overhead.

VI. Conclusions and Future Work

In P2P MPVC systems, using layered coding is the "go-to" method to deal with peer bandwidth heterogeneity. However, it is well-known that layered coders incur significant rate and complexity overheads. Alternatively, one can partition receivers of the same source to multiple groups and distribute single-layer video in each one. In this paper, we have investigated the problem of video quality maximization in P2P MPVC systems for the layered and receiver partitioning systems, under both uplink and downlink capacity constraints. We have shown that any distribution tree can be reduced to a collection of depth-1 and depth-2 trees. Leveraging on this, we have designed an integrated video encoding and distribution algorithm for the layered system. For the receiver partitioning approach, we have formulated the optimal receiver partitioning problem and proposed a simple partitioning algorithm. Our simulations show that the video rates in both systems are very close to the theoretical bounds, but the receiver partitioning system can achieve significantly higher video quality than the layered system, because of the higher coding efficiency of non-layered coding. Leveraging on our theoretic study here, we are developing distributed P2P MPVC protocols for both layered coding and receiver partitioning. The main challenges are adapting to time-varying peer uplink and downlink bandwidth, realtime recovery from packet losses, and control end-to-end video delays experienced by users.

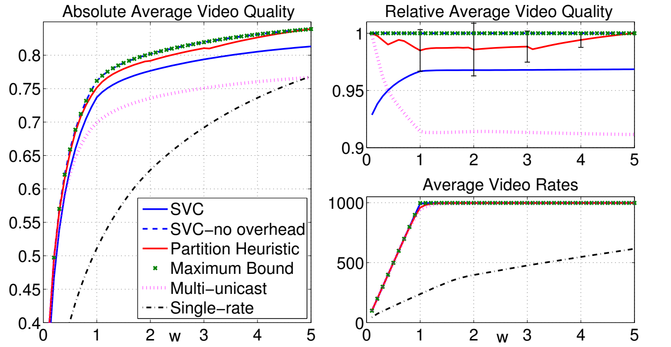

Fig. 3. Comparison of different schemes in a 6-user system, averaged over 50 user bandwidth profiles; absolute quality (left), relative quality normalized to the maximum achievable quality (top right), total throughput (bottom right)

References

[1] Skype, "Video features," http://www.skype.com/.

[2] Google, "Hangouts," http://www.google.com/+/learnmore/hangouts/.

[3] ooVoo, "Homepage," http://www.oovoo.com/.

[4] Y. Xu, C. Yu, J. Li, and Y. Liu, "Video telephony for end-consumers: Measurement study of google+, ichat, and skype," in *ACM Internet Measurement Conference*, 2012.

[5] J. Li, P. Chou, and C. Zhang, "Mutualcast: An efficient mechanism for one-to-many content distribution," in *ACM Sigcomm Asia Workshop*, 2005.

[6] M. Chen, M. Ponec, S. Sengupta, J. Li, and P. Chou, "Utility maximization in peer-to-peer systems," *ACM SIGMETRICS Performance Evaluation Review*, vol. 36, no. 1, pp. 169–180, 2008.

[7] M. Ponec, S. Sengupta, M. Chen, J. Li, and P. A. Chou, "Optimizing multi-rate peer-to-peer video conferencing applications," *IEEE Transactions on Multimedia*, vol. 13, no. 5, pp. 856–868, 2011.

[8] C. Liang, M. Zhao, and Y. Liu, "Optimal bandwidth sharing in multiswarm multiparty p2p video-conferencing systems," *Networking, IEEE/ACM Transactions on*, vol. 19, no. 6, pp. 1704–1716, 2011.

[9] X. Chen, M. Chen, B. Li, Y. Zhao, Y. Wu, and J. Li, "Celerity: a low-delay multi-party conferencing solution," in *Proceedings of the 19th ACM international conference on Multimedia*, 2011.

[10] H. Schwarz, D. Marpe, and T. Wiegand, "Overview of the scalable video coding extension of the h.264/avc standard," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 17, no. 9, 2007.

[11] E. Kurdoglu, Y. Liu, and Y. Wang, "Dealing with user heterogeneity in p2p multiparty video conferencing," *NYU-Poly Technical Report*, 2013, http://eeweb.poly.edu/faculty/yongliu/docs/layvspart-TR.pdf.

[12] H. Hu, Z. Ma, and Y. Wang, "Optimization of spatial, temporal and amplitude resolution for rate-constrained video coding and scalable video adaptation," in *Proceedings of ICIP*. IEEE, 2012.