Optimizing Multi-Rate Peer-to-Peer Video Conferencing Applications

Miroslav Ponec, Sudipta Sengupta, Minghua Chen, Jin Li, and Philip A. Chou

***Abstract*—We consider multi-rate peer-to-peer multiparty video conferencing applications, where different receivers in the same group can receive videos at different rates using, for example, scalable layered coding. The quality of video received by each receiver can be modeled as a concave utility function of the video bitrate. We study and address the unique challenges introduced by maximizing utility in the multi-rate setting as compared to the single-rate case. We first determine an optimal set of tree structures for routing multi-rate content using scalable layered coding. We then develop Primal and Primal-dual based distributed algorithms to maximize aggregate utility of all receivers in all groups by multi-tree routing and show their convergence. These algorithms can be easily implemented and deployed on today's Internet. We have built a prototype video conferencing system to show that this approach converges to optimal bitrates to improve user experience and offers automatic adaptation to network conditions and user preferences.**

***Index Terms*—Multimedia traffic management, quality-of-service, videoconferencing and collaboration environment.**

I. Introduction and Motivation

PROVIDING quality-of-service (QoS) in P2P multiparty conferencing (voice and/or video conferencing) applications is challenging. To maximize the aggregate quality of experience of participating peers, the conferencing system needs to properly allocate the shared network resources, in particular peers' upload bandwidth, and route peers' video streams in an efficient way. The quality of experience of a video conferencing peer is measured by a utility function, which is usually represented by the peak signal-to-noise ratio (PSNR) of the decoded video [2].

There are several existing solutions for conducting P2P multiparty conference. The client-server approach [Fig. 1(a)] ensures that the entire upload bandwidth of each peer can be used for the delivery of just that peer's audio/video stream to the server and the central server (called media control unit—MCU) and then distributes the data to all other peers. However, it places a heavy CPU and network bandwidth burden on the server and thus incurs heavy deployment and routing bandwidth costs. In the ad hoc simulcast approach [Fig. 1(b)], each peer splits its uplink bandwidth capacity equally among all receivers and sends its video to each receiver separately. That is, for an n-party conferencing session, each peer i measures its upload capacity C_i, and sends one conferencing session at coding bit rate $C_i/(n-1)$ to each of the $n-1$ other peers via unicast. Though simple to implement, this approach suffers from poor quality-of-service, especially when there are peers with low upload bandwidth, as those peers are forced to use very low coding rates which degrades the overall experience of the other peers. In recent work [2], overlay routing and allocation of source rates in a P2P multiparty conferencing system is formulated as a multicast optimization problem subject to peer uplink bandwidth constraints. It was shown that the overall system utility can be maximized in a fully distributed manner, by using multiple-trees delivery and running distributed algorithms on participating peers.

However, above solutions assume a single-rate setting, where all receivers of the same multicast group receive content at the same rate. In practice, this assumption does not reflect the possibly diverse needs of peers and available resources. For instance, by using a scalable video codec, sources can generate one video stream that can be decoded at different bit rates. As a result, receivers with larger screens, for example, can receive the video at a higher rate than those with small ones, and get a better experience.

In this paper, we consider the P2P utility maximization problem for a *multi-rate* multicast setting, where different receivers in the same group can receive at different rates. In contrast to the above single-rate case, multi-rate multicast addresses the very diverse needs of peers. Our work is targeted to multiparty video conferencing systems. In such *closed* systems, all participating peers are willing to contribute their upload bandwidth to maximize the aggregate utility, and the number of peers do not go beyond 10–15 most of the time. As such, issues involving peer incentives and scalability to large number of peers are not considered in this paper.

We make the following main contributions:

- **Optimal Tree Packing for P2P Multi-Rate Multicast**: We show that the maximum multicast utility under multi-rate setting can be achieved by routing along a set of depth-1 and depth-2 trees, whose number is *quadratic* in the number of nodes, for each source in the overlay network.
- **Multi-Tree-Based Formulation and Distributed Algorithms**: We provide a new multi-tree-based formulation

Manuscript received September 26, 2010; revised May 16, 2011; accepted June 25, 2011. Date of publication July 14, 2011; date of current version September 16, 2011. A short preliminary version of the paper appeared in IEEE ICME in 2009 [1] . In this version, we have included proofs of all theorems, expanded some sections, and improved presentation. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Yiannis Andreopoulos.

M. Ponec is with the Akamai Technologies GmbH, Media and CDN Engineering Department, Betastrasse 10B, Unterfoehring, Germany (e-mail: mponec@akamai.com).

S. Sengupta, J. Li, and P. A. Chou are with Microsoft Research, Redmond, WA 98052 USA (e-mail: sudipta@microsoft.com; jinl@microsoft.com; pachou@microsoft.com).

M. Chen is with the Department of Information Engineering, The Chinese University of Hong Kong, Shatin, NT, Hong Kong (e-mail: minghua@ie.cuhk.edu.hk).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TMM.2011.2161759

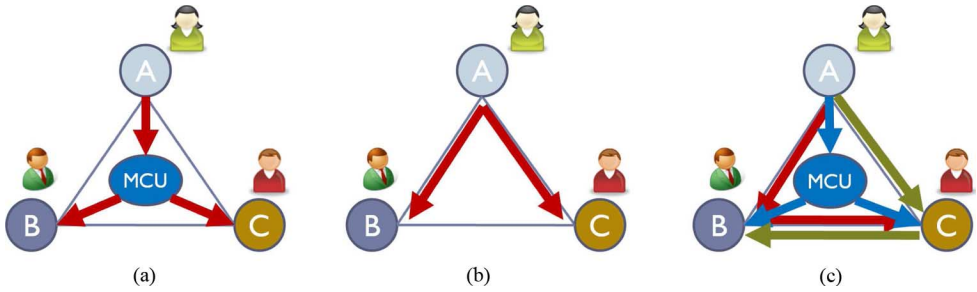


Fig. 1. Different types of content delivery for a three-peer video conference scenario. Delivery of peer A's content to the two other peers, B and C, is shown by arrows. Media control unit (MCU) represents a central server distributing media data of each peer to all other peers (a) MCU-assisted. (b) Simulcast. (c) Peer-assisted.

for P2P multi-rate multicast utility maximization problem, where the variables are rates of individual trees. This is in contrast to the nonlinear constraints in previous formulations using link rates or path rates as variables. We design a packet-marking based Primal and a queuing delay based Primal-dual distributed algorithm, and prove their global asymptotic convergence to optimal solutions of the problem.

- **Virtual Lab Evaluation**: We have implemented a prototype multi-rate multiparty conferencing system using the delay-based Primal-dual algorithm, and evaluated its performance over the Virtual Lab testbed [3]. The results show that the system can achieve the optimal utility as predicted by theoretical analysis. The strict delay requirements for conferencing are also satisfied.

A. Related Work

Utility maximization-based rate control for multicast routing is a well-studied problem (e.g., [4], [5]), though a large body of the work assumes single source, single rate, and single (given) tree settings. Most of these approaches use link rates or path rates as variables, and hence need to handle nonlinear constraints in their formulations. The multi-rate setting for a single source, single tree case has also been considered in [6]. Most of prior related work focuses on underlay networks, and requires additional functionality, such as multicasting and maintaining per-flow states, to be deployed in routers; hence, they are difficult to deploy on today's Internet.

In contrast, we consider the *multi-source multi-rate multicast problem on the overlay network in a P2P setting*, where *routing is performed along a chosen set of trees computed as part of the solution*. Our work focuses on optimal usage of peer uplink bandwidths and ready deployment in the current Internet, and is a multi-rate extension of our previous work on single-rate multiparty conferencing [2]. Using the uplink bottleneck property of P2P topologies, we obtain new formulations and optimality results for multi-rate multicast tree selection in the overlay network and distributed rate control on the trees for utility maximization.

There is also work focused on multicast scenarios where routers can perform intra-session network coding [7]–[9], with and without given multicast trees. The challenge is to deal with non-strictly concave optimization under nonlinear constraints. By exploring the Proximal approach, or a slow timescale traffic engineering control approach, or expressing the constraints involving $\max(\cdot)$ terms with equivalent linear ones, distributed Primal, Dual sub-gradient and Primal-dual algorithms are proposed to maximize the sum of non-concave utility functions, or minimize the cost of using the network [7]–[9]. Chiu *et al.* [10], for example, compare the maximum achievable throughput using network coding with routing in P2P networks and concludes that there is no advantage of using network coding. However, their star network model is very simple and captures only the essential elements of a P2P network in an ideal case with optimal routing. Their model does not apply to our practical conferencing scenarios. We prove that for some common P2P topologies, network coding is not necessary and routing through a linear number of multicast trees per source is sufficient to achieve the same rate region as can be obtained by network coding.

Various types of fairness definitions across users of a P2P system can be warranted by choosing different utility functions [11]. Mo *et al.* [11] describe an end-to-end window-based congestion control, provide a generalization of Kelly's concept of *proportional fairness* [12], and relate the fairness to the optimization problem in order to address the compromise between user fairness and resource utilization. Their algorithm uses information about a propagation delay which is unfortunately unknown to end users. They suggest using the minimum of delays observed so far as an estimate (which is also what TCP Vegas [13] uses). Our implementation described in Section IV uses a similar technique in the queuing delay estimation. The major difference is that we are using one-way delay measurements in contrast to round-trip times. The problem with these queuing delay estimates is that they fail to adapt to the route change when the new route is longer than the original one. A solution which tries to detect re-routing and resets the minimum delay is proposed in [14] and a recent survey of delay-based protocols is available in [15]. One-way delay measurements became recently used by popular P2P software uTorrent [16] in order to prevent congestion and minimize added delay [17].

Based on our framework presented in [1] and [2], Liang *et al.* [18] built a resource sharing solution for multi-swarm multiparty video conferencing systems. Their approach is an extension of the framework to allow multiple (cooperating) video conferences to share bandwidth in an optimal way with respect to a global utility. This is done by either sharing only resources of helper nodes across conferences or by sharing resources among all participating nodes in all conference sessions.

Akkus *et al.* [19] use layered video in a P2P multiparty video conferencing system with the assumption that each peer can

TABLE I
KEY NOTATION

Notation	Definition
N	set of all nodes
E	set of all uplinks of nodes
C_e	capacity of uplink e
S	set of all sources
R_s	set of receivers for source s
x_r^s	receiver r's receiving rate of source s's video
$U_r^s(x_r^s)$	receiver r's utility of receiving source s's video at rate x_r^s
$y_e^{s\ell r}$	flow rate on link e corresponding to ℓ-th layer video from s to r
z_ℓ^s	rate of source s's ℓ-th video layer
G_ℓ^s	set of receivers of source s's ℓ-th video layer
ξ_m	rate of tree m
λ_e	aggregate rate of uplink e

send and receive (at least) one full-quality video stream, i.e., could participate in a one-to-one video conference, and they consider an optimization problem where the number of base layer video receivers is minimized. In contrast to our approach, they utilize periodical round-trip time (RTT) measurements between any two peers and assume the delay in each direction is the same and only depends on the network latency.

Celerity [20] is an approach to multiparty video conferencing with emphasis on low end-to-end delay. It also uses at most depth-2 multicast trees as our approach but removes all paths which would lead to violating a delay bound. The model assumes the capacity bottleneck can be anywhere in the network, unlike our approach where only the uplink bandwidth of peers is considered. They present a packet loss rate-based primal subgradient algorithm to solve a similar optimization problem to ours with an added delay bound.

B. Structure of the Paper

The rest of this paper is organized as follows. We formulate the problem and present the analysis in Section II. We propose distributed algorithms and prove their convergence in Section III. We discuss the practical implementation of our solution in Section IV and evaluate its performance in Section V. Section VI concludes our paper.

II. PROBLEM STATEMENT

In this section, we first explain the video coding model and specifically layered video (Section II-A) and then describe layer settings (Section II-B). Next, we set the optimum with intra-session network coding and formulate the optimization problem (Section II-C). Then we prove that intra-session network coding optimum in a certain topology can be achieved by a particular method of packing multicast trees (Section II-D) and allows reformulation of the optimization problem (Section II-E). In the last part of this section (Section II-F), we focus on a special case when all receivers have the same utility function.

The key notations used in this paper are listed in Table I. We use bold symbols to denote vectors and matrices of these quantities, e.g., $\boldsymbol{x} = \{x_r^s, \forall r \in R_s, \forall s \in S\}$, $\boldsymbol{z^s} = \{z_\ell^s, 0 \le \ell \le |R_s| - 1\}$, and $\boldsymbol{G^s} = \{G_\ell^s, 0 \le \ell \le |R_s| - 1\}$.

A. Video Coding Model

To address the *high variability in the demand for video quality and resources* each peer contributes to the conference session, we use multi-rate multicast, where different receivers may have different demands on the video stream quality and thus may receive different rates of the same video. Scalable video coding can address the very diverse needs of peers. It encodes the content once and then offers the video content as streams of various quality. It is particularly attractive in scenarios where the bandwidth capabilities, system resources, and network conditions are not known in advance.

There are two common approaches for sources to provide multi-rate streams, namely *multiple description coding* (MDC) and *layered coding* . MDC [21], [22] is a coding technique which, instead of generating a single media stream, creates multiple *independent* substreams called descriptions. Receiving any description is enough to decode the video, though receiving more descriptions improves the decoded video quality.

On the contrary, layered coding, used for example in scalable video coding (SVC, or H.264/AVC Annex G) [23], [24], generates a base video layer and several enhancement layers. All receivers need the base layer to successfully decode the video. Enhancement layers can be used to improve the video quality. However, unlike the case of MDC, the layers in layered coding are not independent. The first enhancement layer depends on the base layer, and each subsequent enhancement layer depends on all the lower layers. Such dependence in layers makes layer coding less flexible than MDC. However, layer coding typically has a coding efficiency noticeably higher than that of MDC.

In spite of the benefits scalable video provides, it is not widely adopted today mostly because of the complexity of codec development and decreased compression efficiency compared to single description video coding. However, the availability of good codecs is expanding and so will the popularity of scalable coding; also the compression gap compared to single-layer coding is being minimized.

We use SVC in our approach where both the number of layers each user receives and the layer rates together provide the video quality scalability.

B. Layer Assignment

Suppose for a given source s, the receiver rates are ordered as $x_{i_1}^s \le x_{i_2}^s \le \cdots \le x_{i_{|R_s|}}^s$. We construct $|R_s|$ multicast sessions as follows. The rate $x_{i_1}^s$ can be interpreted as a base layer rate, multicasted from s to all receivers in R_s. The next higher layer, layer 1, has rate $(x_{i_2}^s - x_{i_1}^s)$ and is multicasted from s to all receivers in $R_s - \{i_1\}$. In general, layer ℓ, $0 \le \ell \le |R_s| - 1$ has rate $(x_{i_{\ell+1}}^s - x_{i_\ell}^s)$ and is multicasted from s to all receivers in $\{i_{\ell+1}, i_{\ell+2}, \ldots, i_{|R_s|}\}$. We summarize the above procedure into Algorithm 1.

Algorithm 1: Layer Tate and Session Group Assignment

1. // Input: ordered receiver rates $x_{i_1}^s \le x_{i_2}^s \le \cdots \le x_{i_{|R_s|}}^s$
2. // Output: Layer rate $\boldsymbol{z^s}$ and session group $\boldsymbol{G^s}$
3. $z_0^s = x_{i_1}^s$, $G_0^s = R_s$

4. **for** ℓ from 1 to $|R_s| - 1$**do**

5. $\quad z_\ell^s = x_{i_{\ell+1}}^s - x_{i_\ell}^s$

6. $\quad G_\ell^s = \{i_{\ell+1}, i_{\ell+2}, \ldots, i_{|R_s|}\}$

7. **end for**

8. Output $\boldsymbol{z^s}$ and $\boldsymbol{G^s}$

By transmitting its layered video in these $|R_s|$ multicast sessions, s can deliver its video to $|R_s|$ receivers at the above-mentioned rates.

Note that in a practical implementation of the algorithms in Section III, the layer rates are determined as the sum of rates of all multicast trees delivering the data of each layer. The rates of the multicast trees are determined by these algorithms and are their only output. Network characteristics such as queuing delay together with utility coefficients (described in Section IV) are the input.

C. Rate Region With Intra-Session Coding

For (single- or multi-) session multicast, it is known that network coding, where nodes can mix incoming packets and send out coded packets, can enlarge the achievable multicast rate region as compared to routing [25]. Depending on whether or not packets from different sessions are mixed, we can classify network coding into two types: *inter-session coding* if packets from different sessions are mixed, and *intra-session coding* if only packets from the same session are mixed. It has been shown that nonlinear inter-session coding could give the largest possible rate region [26]; however, computing such mixing and coding is still a largely open problem.

We take the rate region achievable by intra-session network coding, i.e., only packets belonging to the same layer from the same source can be mixed, as the target optimum, and denote the rate region by $\mathcal{B}$. It can be described as follows: $\boldsymbol{x} \in \mathcal{B}$ if and only if for some choice of the routing variables $\{y_e^{s\ell r}, r \in G_\ell^s, 0 \le \ell \le |R_s| - 1, s \in S\}$ the following constraints are satisfied:

Rate Region $\mathcal{B}$ (Intra-Session Coding)

$$\sum_{e \in E^+(i)} y_e^{s\ell r} - \sum_{e \in E^-(i)} y_e^{s\ell r} = \begin{cases} +z_\ell^s, & \text{if } i = s \\ -z_\ell^s, & \text{if } i = r \\ 0, & \text{otherwise .} \end{cases}$$
$$\forall i \in N, r \in \{i_{\ell+1}, \ldots, i_{|R_s|}\},\ 0 \le \ell \le |R_s| - 1, s \in S \quad (1)$$

$$\sum_{s \in S} \sum_{\ell=0}^{|R_s|-1} \max_{r \in R_s} (y_e^{s\ell r}) \le C_e \ \forall e \in E$$

$$z_0^s = \min_{r \in R_s} (x_r^s) \ \forall s \in S$$
$$z_\ell^s = \min_{r \in R_s}^{\ell+1} (x_r^s) - \min_{r \in R_s}^{\ell} (x_r^s)$$
$$\forall 1 \le \ell \le |R_s| - 1, s \in S \quad (2)$$

where $E^-(i)$ denotes the links going into node i and $E^+(i)$ links leaving node i, and $\min^\ell$ to denote the ℓth minimum of a set of numbers (e.g., $\min^1$ is the usual minimum).

The constraints in (1) are the flow balance constraints. That is, for any node i other than source s and receiver r, the amount of outgoing traffic must be equal to the amount of incoming traffic. For source s and receiver r, the difference between these two traffic amounts must be equal to the ℓth video layer rate. The constraints in (2) are the upload capacity constraints. That is, for uplink $e \in E$, the amount of outgoing traffic across all sessions must be less than its uplink capacity C_e. The max term models the coding within a session. Over the convex region $\mathcal{B}$, the multi-rate multicast utility maximization problem can be stated as follows:

1) Problem 1 (Multi-Rate Multicast Utility Maximization):

$$\max_{\boldsymbol{x}} \quad \sum_{s \in S} \sum_{r \in R_s} U_r^s(x_r^s), \quad \text{s.t.} \quad \boldsymbol{x} \in \mathcal{B}. \qquad (3)$$

D. Achieving Rate Region $\mathcal{B}$ in P2P Topology

We now consider how the rate region $\mathcal{B}$ can be achieved. In the widely accepted P2P topology model [2], [27], *peer uplinks are the only bottlenecks in the network, and every peer can directly connect to every other peer through routing in the underlay.*

Under this model, a powerful theorem established in [27] states the following. Consider a *single-rate single-source* multicast scenario over a P2P network, with the source s, a set of receivers R_s, and a set of helpers H. A helper is neither source nor receiver, but an intermediate node which receives data from source and distributes it to receivers. Then, the rate region achieved by intra-session network coding can also be achieved by packing at most $1+|R_s|+|H|$ multicast trees as follows: 1) One depth-1 tree rooted at s and reaching all receivers in R_s, 2) $|R_s|$ depth-2 trees, each rooted at s and reaching all other receivers in R_s via different $r \in R_s$, and 3) $|H|$ depth-2 trees, each rooted at s and reaching all receivers in R_s via different $h \in H$. Notice that this result is valid for the *single-rate single-source* multicast scenario. It has been recently extended to the *multi-source single-rate* multicast scenario [2].

We now extend the above result to the *multi-source multi-rate* scenario, for which we need to extend the depth-1 and depth-2 tree definitions to allow subsets of receivers as follows:

- **Depth-1 type tree**: Rooted at a given source s and reaching a *subset* of receivers in R_s through direct link(s) from s.
- **Depth-2 type tree**: Rooted at a given source s, reaching a receiver $r \in R_s$ or helper $h \in H$ through a direct link from s, and from the latter node reaching a *subset* of receivers in R_s through direct link(s).

An example of these two types of trees for a single source are shown in Fig. 2. Fig. 3 then shows all multicast trees of the types described previously for two conferences, one with three peers and a helper, and one conference with four peers, respectively.

Suppose we know the ordering of receiver rates $x_r^s, r \in R_s$ for each source $s \in S$, and denote this ordering by $\pi = (\pi^s, s \in S)$, where π^s is a permutation of the receivers $r \in R_s$. The number of such different π is $\prod_{s \in S} |R_s|!$ Note that the number is not important in practice because we always work with a specific π. We use π_i^s to denote the ith receiver in the permutation order for source s. Let $\mathcal{B}(\pi)$ be the subset of rate region $\mathcal{B}$ where the receiver rates x_r^s for any given source s are ordered according to π. We first establish that the rate region $\mathcal{B}(\pi)$, achieved by intra-session network coding, can also be achieved by routing.

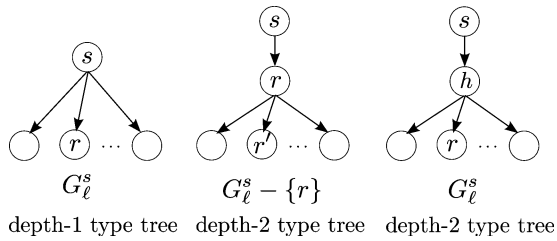


Fig. 2. Depth-1 type and depth-2 type multicast trees. Here G^s_ℓ serves as an example subset of R_s.

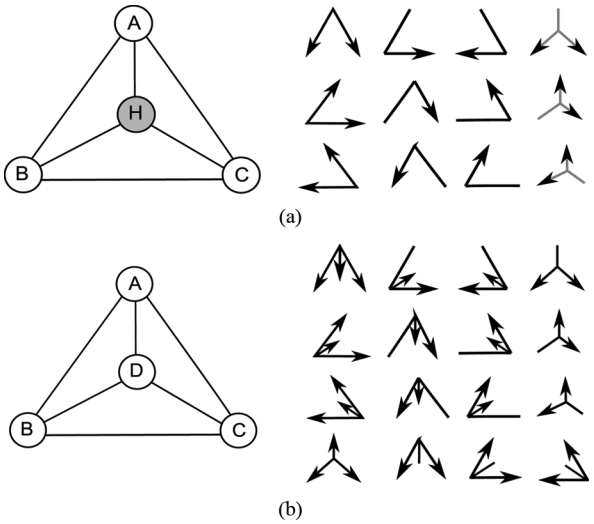


Fig. 3. Depth-1 type and depth-2 type multicast trees for all sources for two different conference scenarios: (a) 3 peers (A, B, C) and one helper (H) and (b) 4 peers (A, B, C, D).

Theorem 1: The rate region $\mathcal{B}(\pi)$ can be achieved by packing depth-1 type and depth-2 type trees. The tree construction procedure for a source s is given in Algorithm 2.

Proof: Consider a given source $s \in S$. For this source, the receiver rates are ordered as $x^s_{\pi^s_1} \le x^s_{\pi^s_2} \le \cdots \le x^s_{\pi^s_{|R_s|}}$. Using Algorithm 1, we construct layered video sessions $\boldsymbol{G}^s$ and the corresponding rates $\boldsymbol{z}^s$ for source s.

Since for rate region $\mathcal{B}$, there is no network coding across layer sessions. Each layer can then be considered an independent session and the theorem can be applied to each layer. To achieve $\mathcal{B}(\pi)$, it is then sufficient for every source s to route its layer ℓ's video to receivers in $G^s_\ell (0 \le \ell \le |R_s| - 1)$, using one depth-1 type tree reaching all receivers in G^s_ℓ from s, and G^s_ℓ depth-2 type trees reaching one receiver in G^s_ℓ from s and then the rest of receivers in G^s_ℓ. ■

We summarize the above tree construction process for a source s in Algorithm 2.

Algorithm 2: Layer Trees Construction

1. // Input: Session group $\boldsymbol{G^s}$ of source s

2. // Function: Construct depth-1 type and depth-2 type trees to deliver s's layered video

3. **for** ℓ from 0 to $|R_s| - 1$ **do**

4. Construct a depth-1 type tree reaching all receivers in G^s_ℓ from s

5. **for** $r \in G^s_\ell$ **do**

6. Construct one depth-2 type tree reaching r from s, and then to the rest of receivers in $G^s_\ell - \{r\}$

7. **end for**

8. **end for**

The lemma below states that certain trees need not be considered when distributing the layers for a given source. In particular, for layer ℓ, these are the depth-2 type trees that use a helper which is a receiver of a lower layer but not of layer ℓ.

Lemma 1: In an *optimal solution* for the rate region $\mathcal{B}(\pi)$, for each source $s \in S$, node π^s_ℓ (for any $0 \le j \le |R_s| - 2$) will not be a helper in the depth-2 type trees considered for layers $(\ell + 1)$ and higher.

Proof: We prove this by contradiction. Suppose the above is not true. Then, the additional rate received by node π^s_ℓ as a helper in any of the depth-2 type trees in higher layers can be used to assign it a higher rate. This might change the position of node π^s_ℓ in the resulting ordering, but it can be verified that the new set of rates is still feasible. Since this increases the utility received by node π^s_ℓ, it contradicts the optimality of the solution. ■

Note that non-receiver nodes for source s can participate as helpers for depth-2 type trees for this source. Thus, the number of trees used to distribute layer 0 for source s is at most $1 + |R_s| + (|N| - |R_s| - 1) = |N|$. Using the above lemma, the total number of trees that need to be considered for routing data from source s in order to achieve the rate region $\mathcal{B}(\pi)$ for any given π is

$$\sum_{\ell=0}^{|R_s|-1} (|N| - \ell) = |N||R_s| - \frac{|R_s|(|R_s| - 1)}{2} - 1 \quad (4)$$

which is at most *quadratic* in the total number of peer nodes in the network.

Since receivers' rates for the same source may be different in the multi-rate multicast problem, we cannot directly use the multi-source single-rate multicast result in [2] to restrict the number of trees to be considered in order to achieve the rate region $\mathcal{B}$. The theorem below establishes that the optimal solution in $\mathcal{B}$ can indeed be expressed as a linear superposition of flows along depth-1 and depth-2 type trees.

Theorem 2: The optimal solution in rate region $\mathcal{B}$ can be expressed as a linear superposition of flows along depth-1 type and depth-2 type trees for every source s in S.

Proof: Since the rate region $\mathcal{B}$ can be expressed as a union of rate regions $\mathcal{B}(\pi)$ over all π, we have $\mathcal{B} = \cup_\pi \mathcal{B}(\pi)$.

From Theorem 1, we know that the optimal solution in each rate region $\mathcal{B}(\pi)$ can be expressed as a linear superposition of flows along depth-1 type and depth-2 type trees. By taking the best solution among the optimal solutions for each rate region $\mathcal{B}(\pi)$, we obtain the optimal solution for rate region $\mathcal{B}$. This establishes the result. ■

E. Tree-Based Formulation for P2P Multi-Rate Multicast Utility Maximization

Receiver nodes on a tree m receive the same content at the same rate ξ_m. With a slight abuse of notation, we also denote by s the set of trees rooted at source s. Let the aggregate rate of link e be λ_e, i.e., the sum of the rates of tree branches passing through e, and is given by

$$\lambda_e = \sum_{s \in S} \sum_{m: m \in s, e \in m} b_e^m \xi_m, \ \forall e \in E \tag{5}$$

where b_e^m is the number of branches of tree m that pass through physical uplink e. Since different branches of a tree emanating out of the same node pass through the same physical uplink, the tree rate may be counted multiple times when computing the aggregate rate of link e, hence the multiplication by b_e^m. Based on Theorem 2, we reformulate Problem 1 as follows:

1) Problem 2 (Tree-Based Multi-Rate Utility Maximization):

$$\begin{aligned} \max_{\boldsymbol{\xi}} \quad & \sum_{s \in S} \sum_{r \in R_s} U_r^s \left(\sum_{m: m \in s, r \in m} \xi_m \right) \\ \text{s.t.} \quad & \lambda_e \le C_e, \ \forall e \in E. \end{aligned} \tag{6}$$

Remarks: This tree-based formulation avoids the $\max$ term in (2) that is present in a link flow based formulation as in Problem 1. Moreover, by using flows on trees as variables, our solution explicitly takes routing of sub-streams into account and facilitates a distributed rate control-based solution. In multiparty conferencing scenarios, the utility function itself can depend on the nature of the video source and the receivers. For example, a video for a source with heavy motion will have a utility function that offers steep increase for a large bit rate range, and may thus favor a higher coding bit rate, while the source with low motion will have a utility function that flattens quickly with the increase of the coding rate, and may thus be satisfied with a relatively low coding bit rate. Also, the utility function depends on receivers' preference, for example, the display resolution with which they are currently viewing each video from each source.

F. Case of Receiver-Independent Utility Functions

In a special case when all receivers for the same source have the same utility function, it can be expected that an optimal solution allocates equal rates to all receivers for the same source. This is, in fact, true and is established in the following theorem.

Theorem 3: For rate region $\mathcal{B}$, if $U_r^s = U^s \ \forall r \in R_s, s \in S$ (receiver utility functions are identical for the same source), then there exists an optimal solution in which $x_r^s = x^s \ \forall r \in R_s, s \in S$ (receiver rates are identical for the same source).

Proof: Let $\{x^{*s}{}_r\}$ denote an optimal solution for the problem in rate region B in which the receiver rates for the same source are not all equal for at least one source. We will show that there exists a feasible solution in which

$$x_r^s = x^s = \frac{\sum\limits_{r \in R_s} x^{*s}{}_r}{|R_s|} \ \forall \, r \in R_s, s \in S.$$

Then, using Jensen's inequality for concave functions, we have

$$\sum_{s \in S} \sum_{r \in R_s} U^s(x^s) \ge \sum_{s \in S} \sum_{r \in R_s} U^s(x^{*s}{}_r). \tag{7}$$

This shows that $\{x_r^s\}$ is a feasible solution with objective function value at least (and hence equal to) that of the optimal solution $\{x^{*s}{}_r\}$, thus proving the theorem.

It remains to show the feasibility of the rates $\{x_r^s\}$. To establish this, we start from the feasible solution $\{x^{*s}{}_r\}$ and show that receiver rates for each source can be adjusted to be equal while preserving the feasibility of the solution.

Consider source s and let the receiver rates in the given optimal solution be ordered as $x^{*s}{}_i \le x^{*s}_{i_2} \le \cdots \le x^{*s}_{i_{|R_S|}}$.

Each receiver in R_s receives layer 1 (the base layer) with rate x_i^{*s}. Layer 2 is received by each receiver in $R_s - \{i_1\}$ at rate $\Delta^{s_2} = x_{i2}^{*s} - x_{i1}^{*s}$. The distribution of layer 2 for source s can be decomposed into flows along depth-1 and depth-2 trees as follows. Let α_0 be the rate associated with the depth-1 tree T_0 rooted at s and reaching all receivers in $R_s - \{i_1\}$ directly. Let α_r be the rate associated with the depth-2 tree T_r rooted at s and using receiver $r \in R_s - \{i_1\}$ as helper. Let β_h be the rate associated with the depth-2 tree T_h rooted at s and using non-receiver node h as helper. Then, the rate for layer 2 is given by $\Delta_2^s = \alpha_0 + \sum_r \alpha_r + \sum_h \beta_h$.

We now add receiver node i_1 to upload from the helper node in each of the trees T_r and T_h, so as to obtain trees T_r' and T_h', respectively. Without using additional uplink bandwidth at the helper node of each depth-2 tree, tree T_r' can now support a rate of $\alpha_r(|R_s|-2)/|R_s|-1$ to all receivers in R_s and tree T_h' can now support a rate of $\beta_h(|R_s|-1)/|R_s|$ to all receivers in R_s. This frees up uplink capacity of $(\sum_r \alpha_r/|R_s| - 1 + \sum_h \beta_h/|R_s|)$ at source s.

We now remove tree T_0 from the solution, thus freeing up additional uplink capacity of $(|R_s|-1)\alpha_0$ at source s. The total freed up uplink capacity at source s is now used to serve all receivers in R_s through a depth-1 tree T_0 with rate $1/|R_s|(\sum_r \alpha_r/|R_s| - 1 + \sum_h \beta_h/|R_s| + (|R_s|-1)\alpha_0)$.

We have now redistributed the flow for layer 2 (subject to node uplink constraints) so as to reach all receivers in R_s with equal rates of

$$\begin{aligned} & \frac{\alpha_r(|R_s|-2)}{|R_s|-1} + \frac{\beta_h(|R_s|-1)}{|R_s|} \\ & \quad + \frac{1}{|R_s|}\left(\sum_r \frac{\alpha_r}{|R_s|-1} + \sum_h \frac{\beta_h}{|R_s|} + (|R_s|-1)\alpha_0 \right) \\ & \quad = \frac{(|R_s|-1)}{|R_s|}(\alpha_0 + \sum_r \alpha_r) + \left(\frac{|R_s|-1}{|R_s|} + \frac{1}{|R_s|^2} \right) \sum_h \beta_h \\ & \quad \ge \frac{(|R_s|-1)}{|R_s|}\left(\alpha_0 + \sum_r \alpha_r + \sum_h \beta_h \right) = \frac{(|R_s|-1)}{|R_s|}\Delta^{s_2}. \end{aligned}$$

Hence, with receiver i_1 added to the receivers for layer 2, the new rate reaching all receivers in R_s decreases by a factor of $(|R_s|-1)/|R_s|$ relative to the earlier rate Δ_2^s reaching all receivers in $R_s - \{i_1\}$, while the total served by source s to all receivers in layer 2 remains the same.

We can repeat this procedure similarly for layers 3 and higher so that flows for each layer are distributed to all receivers in R_s (subject to node uplink constraints). Since the total rate served by source s remains the same at every step of the procedure, the final rate received by each receiver in R_s is clearly $1/|R_s|\sum_{r\in R_s} x^{*s}{}_r = x^s$. Hence, the rates $\{x_r^s\}$ are a feasible solution for the problem. This completes the proof. ■

If the utility functions U^s are *strictly* concave, then it can be shown that *any* optimal solution will have equal receiver rates for the same source. We summarize this observation as follows.

Corollary 1: For rate region $\mathcal{B}$, if $U_r^s = U^s\ \forall r \in R_s, s \in S$ and the utility functions U^s are *strictly* concave, then in every optimal solution, we have $x_r^s = x^s\ \forall r \in R_s, s \in S$.

III. Distributed Algorithms

A. Packet Marking-Based Primal Algorithm

The Primal algorithm follows the penalty approach by relaxing the constraints by adding a penalty to the objective function whenever constraints are violated. In particular, we study the following penalty version of the problem:

$$\max_{\boldsymbol{\xi}} \sum_{s\in S}\sum_{r\in R_s} U_r^s\left(\sum_{m:m\in s, r\in m}\xi_m\right) - \sum_{e\in E}\int_0^{\lambda_e} q_e(w)\,dw \quad (8)$$

where $\int_0^{\lambda_e} q_e(w)\,dw$ is the price associated with violating the capacity constraint of uplink e.

If $q_e(\cdot)$ is non-decreasing, continuous, and not always zero, then the above optimization problem is concave and has at least one equilibrium [12]. The strict concavity of $U_r^s(\cdot)$ indicates that x is unique for any optimal solution. If $-\int_0^{\lambda_e} q_e(w)\,dw$ is also strictly concave, then $\lambda_e, e \in E$, are also unique.

We choose $q_e(w) = (w - C_e)^+/w$ for link e. In terms of ECN marking [28], it represents the packet marking probability.

We consider the following Primal algorithm: $\forall s \in S, \forall m \in s$

$$\dot{\xi}_m = f_m(\xi_m)\left(\sum_{r\in m} U_r'^s\left(\sum_{m:m\in s, r\in m}\xi_m\right) - \sum_{e\in m} b_e^m q_e(\lambda_e)\right) \quad (9)$$

where $f_m(\xi_m)$ is a positive function adjusting the rate of adaptation for ξ_m, and can be chosen arbitrarily.

It can be shown that trajectories of the above system globally asymptotically converge to one of its equilibria, by using La Salle principle, and following the classical arguments by Kelly *et al.* [12]. Moreover, it is also possible to show that the convergence is actually semi-globally exponentially fast, by using a readily available lemma in [29].

B. Queuing Delay-Based Primal-Dual Algorithm

Another way to solve the concave optimization problem in a distributed manner is to look at its Lagrangian:

$$L(\boldsymbol{\xi}, \boldsymbol{p}) = \sum_{s\in S}\sum_{r\in R_s} U_r^s\left(\sum_{m:m\in s, r\in m}\xi_m\right) - \sum_{e\in E} p_e(\lambda_e - C_e) \quad (10)$$

where p_e is the price of using uplink e. There is no duality gap, since the original problem is a concave optimization problem with linear constraints, and strong duality holds.

As a result, any optimal solution of the original problem and its corresponding Lagrangian multiplier forms a saddle point of L over the set $\{\boldsymbol{\xi} \geq 0, \boldsymbol{p} \geq 0\}$, and any saddle point of L gives an optimal solution. It is known that (ξ, p) is a saddle point of L if and only if it satisfies the Karush-Kuhn-Tucker conditions: $\forall s \in S, \forall m \in s, \forall e \in E$

$$p_e \geq 0, \lambda_e \leq C_e, p_e(\lambda_e - C_e) = 0 \quad (11)$$

$$\sum_{r\in m} U_r'^s\left(\sum_{m:m\in s, r\in m}\xi_m\right) - \sum_{e\in m} b_e^m p_e = 0. \quad (12)$$

The optimal Lagrangian multiplier can be nonzero only if the capacity constraint of link e is activated, i.e., $\lambda_e = C_e$.

There could be multiple saddle points of L since the objective function in the original optimization problem in (2) is not strictly concave. We consider the following Primal-dual algorithm to pursue one of the saddle points, over the set $\{\xi \geq 0, p \geq 0\}$: $\forall s \in S, \forall m \in s$, and $e \in E$

$$\dot{\xi}_m = k_m\left(\sum_{r\in m} U_r'^s\left(\sum_{m:m\in s, r\in m}\xi_m\right) - \sum_{e\in m} b_e^m p_e\right)_{\xi_m}^+ \quad (13)$$

$$\dot{p}_e = \frac{1}{C_e}(\lambda_e - C_e)_{p_e}^+ \quad (14)$$

where k_m is a positive constant controlling the adaptation rate of tree m and $(a)_b^+ = a$ if $b > 0$, and is $\max(0, a)$ otherwise. It is known that p_e adapted according to (14) can be interpreted as queuing delay [30] on uplink e.

In [31], the authors show that trajectories of the above Primal-dual system globally converge to the equilibrium, under single-path unicast setting. This result, however, does not apply to our problem since we are under multi-tree multicast setting—there are multiple receivers for every source, and there are multiple paths between the source and any of its receivers.

Under multi-tree/multipath delivery setting, it is shown that the queuing delay following (14) can oscillate indefinitely and may never converge [32, Section 2.5]. In our previous work in [2], we give a sufficient condition for the Primal-dual system in (13)–(14) to converge to the equilibria, and use it to show the convergence of the Primal-dual system in P2P *single-rate* multicast scenario. However, the result does not directly apply to the P2P *multi-rate* multicast scenario.

In the following theorem, we show trajectories of the Primal-dual system in fact converge to the equilibria, in the P2P *multi-rate* multiparty conferencing scenario. The key is to utilize the unique structure of the multicast trees used in our solution, and the fact that peer uplinks are the only bottleneck in the network to verify that the sufficient condition proposed in [2] is satisfied.

Theorem 4: For P2P multi-rate multiparty conferencing scenario, all trajectories of the system in (13)–(14) converge to one of its equilibria globally asymptotically, if k_m are the same for all the trees $m \in s$.

Proof: We utilize [2, Theorem 1] for the proof. This theorem gives a sufficient condition for the Primal-dual algorithm in (13)–(14) to converge to a saddle point of L in (10). We show that the sufficient condition is satisfied in our P2P multi-rate multicast case.

Let A be the connectivity matrix, where the (i, j) entry is the number of branches of tree j passing through link i. Note that

this is different from connectivity matrices in unicast scenarios, as the entries of A now can take nonnegative values other than 1 or 0. Let $K = \text{diag}\{k_m, m \in s, s \in S\}, C = \text{diag}\{C_j, j \in J\}$, where J is assumed to contain only the bottlenecks without loss of generality. Let B be the matrix representing relation between the receiving rates and the tree rates, with the (i, j) entry being 1 if receiver i is on tree j and 0 otherwise.

In the case of multi-rate multiparty scenario, B can be expressed as follows:

$$B = \begin{bmatrix} B_b \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ B_o \end{bmatrix}$$

where B_b contains $|S|$ rows corresponding to S peers each receiving only base layer contents from a unique source, and B_o contains other rows. Clearly, such B_b always exists in multi-rate multiparty conferencing scenario.

To satisfy the condition shown in [2, Theorem 1], it is sufficient to show rank $(BKA^T) = |J|$. In multi-rate multiparty conferencing, we have $|J| = |S|$ since every user is a receiver. Moreover, if $k_m = k_s, m \in s$ take the same value k_s for the same source s, then $BKA^T = K_S BA^T$, where $K_S = \text{diag}\{k_s, s \in S\}$; hence, we have

$$BKA^T = K_S BA^T = K_S \left(\begin{bmatrix} B_b \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ B_o \end{bmatrix} \right) A^T.$$

Consequently, it is sufficient to show $K_S B_b A^T$ has rank $|S|$. Writing out A and working out the math, we have $K_S B_b A^T = K_S \left(|S| I + (|S| - 1) D \right)$, where D is a $|S| \times |S|$ matrix with every entry being 1. $(|S| I + (|S| - 1) D)$ is positive definite matrix since it is the sum of one positive definite matrix and one positive semi-definite matrix. As K_S is a positive diagonal matrix, $K_S \left(|S| I + (|S| - 1) D \right)$ has rank n. Hence, the system in (13)–(14) for multi-rate multiparty conferencing converges to its equilibria, which are the saddle points of L. ■

The Primal-dual algorithm described in (13)–(14) can be implemented by each link generating its queuing delay and each source adjusting the rates of its trees by collecting incentives to increase the tree rates from different receivers, i.e., the derivative of their utility functions, and sum of the queuing delays introduced by using the trees. The algorithm is suitable for implementation in a distributed manner in today's Internet and is discussed further in Section IV.

IV. Practical Implementation

We have implemented the queuing delay-based distributed algorithm (13)–(14) in a prototype of a P2P multi-rate multiparty video conferencing system.[1]

In this system, each peer is a source of its video stream and wants to receive videos from all other peers. Besides encoding and decoding video streams, every peer builds a set of trees used to deliver its video stream and updates them upon peers joining and leaving. The peer is also responsible for controlling the flow rates of this set of trees according to (13), based on the measured queuing delays it collected from other peers.

[1]Please note that we consider only the transmission of video streams in this paper. We assume audio streams are encoded and treated separately. Audio stream bit rates today are also typically much smaller than video bit rates. It may be possible to follow a similar approach presented here for video streams and apply it to audio, but we have not pursued this option. In practice, depending on the scale of the conference, audio streams can be transmitted to a central server, dedicated to audio stream distribution, and then delivered to all peers.

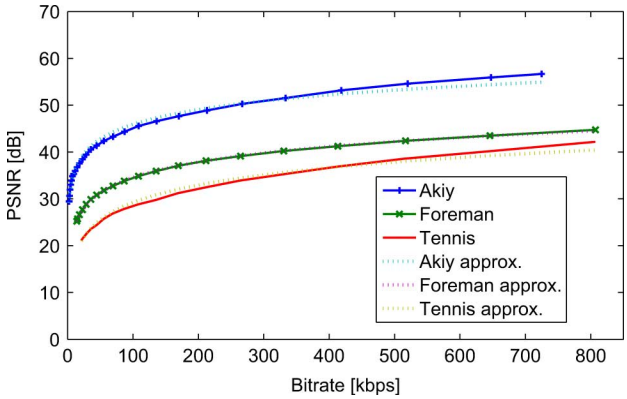


Fig. 4. Logarithmic approximation of PSNR curves of Akiy, Foreman ,and Tennis video sequence, cited from [2].

All multicast trees in our system have depth at most two; hence, a packet traverses at most one overlay hop before reaching its destination. This is important for keeping the *total end-to-end delivery delay low*, thus satisfying the strict requirements of real-time multiparty conferencing systems.

A. Utility Modeling and Layer Assignment

PSNR is the de facto standard metric in video processing to provide objective quality evaluation between the original frame and the compressed one. As shown in Fig. 4, we empirically found that the PSNR of a source s's video coded at rate z_s can be approximated by a logarithmic function $\beta_s \log(z_s)$, with large β_s for videos with large amount of motion and small β_s for almost still videos. This parameter β_s, called *source utility coefficient*, can be obtained from the video encoder during encoding process.

In our implementation, when a peer r subscribes to a video stream of source s, it submits a *receiver utility coefficient*, denoted by β_r^s, to the source. The coefficient β_r^s takes value between 0 and 1, and corresponds to peer r's preference on receiving high-quality video. The smaller the β_r^s, the lower desire for high-quality video receiver r has. Using β_r^s, the source reconstructs receiver r's utility as $\beta_s \beta_r^s \log(x_r^s)$. The aggregate utility the conferencing system optimizes is then given by

$$\sum_{s \in S} \sum_{r \in R_s} \beta_s \beta_r^s \log \left(\sum_{m : m \in s, r \in m} \xi_m \right)$$

and is concave.

Source s sorts all receivers according to their receiver utility coefficients, assigns layers to receivers as described in Section II-B, and builds the set of trees to distribute these layers of video according to Algorithm 2. By doing so, the system intends to solve the multi-rate problem in (6) by using this set of trees.

Note the obtained optimal solution to the problem in (6) is not necessarily the optimal solution to the problem in (3), depending on whether the set of trees the system constructs according to receivers' preferences is optimal. In the case where the order of the receiving rates matches the order of the preferences (β_r^s),

the set of trees the system constructs is optimal and the system solves both problems in (3) and (6).

The final quality, i.e., video bitrate, at which each receiver views the video of each source is based on the output of the algorithms in Section III. There is no way for a receiver to choose a target bitrate they wish to watch directly because the optimal distribution of rates among peers is based on dynamic conditions and is part of the solution. The input receivers provide is their video quality preference and the output of the algorithm is the optimal distribution of rates that respects these preferences and current network and video characteristics. The preferences (represented by β_r^s coefficients) are determined by the conferencing application based on, for example, the device's screen resolution, ability to decode the video at high quality, playback history, and user preference. The values of the coefficients have a relative meaning and even a low-quality preference can lead to receiving a high bitrate video assuming the choice of other peers' coefficients and current network conditions allow it.

B. Queuing Delay Measurement

We use the difference in the *relative one-way-delay* (ROWD) to measure the queuing delay between two peers. ROWD is the relative difference between the packet sending time at the sender peer, and the packet receiving time at the receiver peer. It is the sum of propagation delay, queuing delay, and clock offset between the two peers. It is known that queuing delay p_e between two peers can be estimated by the difference between current ROWD and the smallest ROWD ever seen for this peer. The advantage of measuring delay based on ROWD is that it does not require any time synchronization across peers.

Note that the queuing delay typically includes also the queuing delay in the network, not just the queuing delay introduced at the sender, and because peer uplinks are assumed to be the only bottlenecks in the network, the network queuing delay is assumed to be zero in this paper.

In particular, we follow the following procedure to measure the queuing delays of peers' uplinks and to distribute them among peers.

- Whenever a peer sends or forwards a packet, a timestamp is attached to it.
- Each of its offspring peers on the tree computes the current ROWD, subtracts the minimum ROWD observed so far for the sending peer, and generates a measurement of the queuing delay of the sender's uplink.
- Each peer periodically sends the aggregated queuing delay measurements to all sources to which it is subscribed. If the target source is also a receiver of the peer's video stream, the queuing delay measurements can be piggybacked to its next video packet which guarantees their distribution.

The overhead of distributing the delay information is negligible as it only requires few bytes per packet received and a lot of the overhead can be saved by aggregating the measurements into fewer packets and by piggybacking the delay information to video packets.

Such method is well adopted in the context of measuring congestion [33]. We use it in this paper for a different purpose of measuring queuing delay. The disadvantage of using ROWD to measure the queuing delay is that its measurement can be inaccurate if the underlying route between peers changes; there have been efforts to overcome this drawback [33]. An inconsistent active queue management (AQM) performed by network appliances, i.e., inconsistently prioritizing certain packets over others, could lead to artificial changes to the measured delays and could negatively impact the convergence of these algorithms. Wide-scale problems with AQM have not been reported in practice and more details can be found in [17].

Upon collecting p_e $(e \in E)$, source peer s computes an average queuing delay for each peer on its trees, by doing a running average over the last three queuing delay measurements for the peer. The purpose of doing so is to achieve a balance between robustness to measurement noise and quick response to network condition changes. Source s then updates its tree rates according to (13).

C. Performance, Scalability, and Protocol Overhead

The presented algorithms have low demands on computational power and state maintained, and are negligible when compared to the encoding and decoding processes of multiple video streams happening in a video conference session by each peer. Note that in the video conference, all participants are viewing all other participants' streams, and therefore, conferences of large numbers of participants are not common today. The number of multicast trees each source has to consider is quadratic in the number of participants, and our approach is thus suitable for these small-scale video conferences. For a large video conference or a conference where there are only few sources of videos and many receivers, this approach would not be practical.

Note that peers in our approach do not perform any packet coding on top of standard video streams encoding and decoding. All video packets are forwarded without any changes or processing of the video data, which is important for keeping fast end-to-end delivery times in a latency-sensitive application such as video conference. Moreover, we proved that in a P2P topology, our approach achieves the same rate region as can be achieved by using computationaly intensive network coding. Our straightforward prototype implementation shows there is no resource intensive processing required throughout the system.

Every video stream of each participant in the conference has to reach every other participant and therefore the distribution of the measured queuing delays, which is the only dynamic input in the algorithm (III-B) together with video source characteristics, is easily performed by appending it to the video data being disseminated. This way all peers have up-to-date view of the state of the network overlay. Also the overhead of the protocol on the network is therefore very low as described in the previous section. The rate allocation algorithm works well as long as there are enough packets sent by each peer because each packet provides a queuing delay measurement. Note that every peer participating in the conference keeps utilizing all of their available bandwidth to send video packets of its video or forwarding packets of other peer's video stream and the requirement is therefore trivially satisfied. Every peer keeps sending enough packets and thus providing queuing delay measurement data as long as they are active in the conference. The size of the

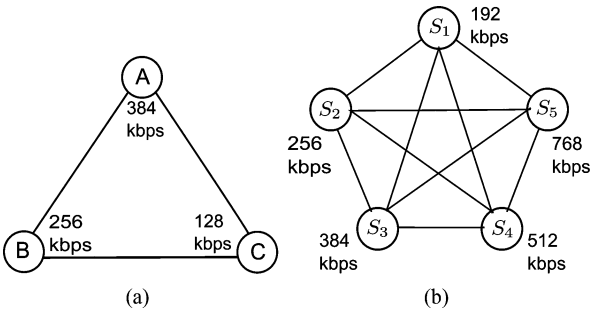


Fig. 5. (a) Topology of Scenario 1 and peer uplink bandwidth setting. (b) Topology of Scenario 2 and peer uplink bandwidth setting.

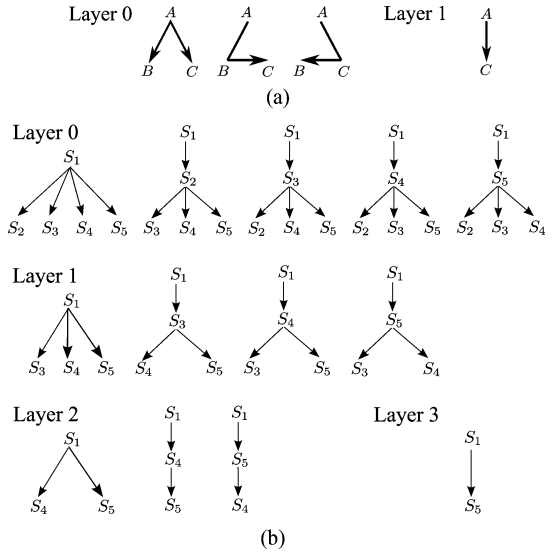


Fig. 6. (a) Multicast trees delivering data of video layers of source A in a three-party conference in Scenario 1. (b) Multicast trees delivering data of video layers of source S_1 in a five-party conference in Scenario 2.

packets matters because the larger the packets the less measurements are available. In practice, this is not a problem because if a peer has small uplink capacity, its video bit rate would be small, too, and the basic requirement on the video data which has to be sent ensures there will be enough small packets used. A peer with large uplink capacity will be sending enough packets regardless of their size (assuming standard MTU of about 1500 bytes on an Ethernet network or smaller). Our approach therefore matches the requirements of any common video conference application and fully utilizes all of the already satisfied conditions.

V. Experimental Results

We use a set of virtual machines in a Virtual Lab infrastructure [3] to conduct experiments in Scenarios 1 and 2 to evaluate the performance of our multiparty conferencing prototype described in Section IV.

A. Scenario 1: The Case of Cross Traffic, Utility Change, and Receiver-Independent Utility Function

The first scenario that we study consists of three peers A, B, and C. The topology and peer uplink bandwidth are shown in Fig. 5(a), from which we can see peer C has the smallest uplink bandwidth. The propagation delays between any two peers are set to be 20 ms.

We study the case where all receivers of a source have the same utility functions, i.e., the receiver-independent utility case. For this, we set all receiver utility coefficients to be 1. Consequently, receiver r of s, where $s, r \in \{A, B, C\}$ and $s \neq r$, has a utility function $\beta_s \log(x_r^s)$ according to our utility modeling in Section IV-A. The aggregate utility our multi-rate conferencing system tries to maximize is $\sum_{s,r\in\{A,B,C\},s\neq r} \beta_s \log(x_r^s)$.

In this scenario, each peer encodes its video into two layers: a base layer and an enhancement layer. Each layer's video is sent along a set of depth-1 and depth-2 trees which are constructed according to the procedure in Section II-C. For instance, as shown in Fig. 6(a), peer A uses three trees to send its base layer video, and uses one tree for its enhancement layer video.

We also evaluate how the system adapts to cross traffic and source utility coefficient changes in this experiment. Initially the conference starts with $\beta_A = \beta_B = \beta_C$. At the 240th second, β_B is increased by 30% as the motion characteristics of the video of user B changes, e.g., the participant starts moving a lot. After another 240 seconds, peer A starts some other application which consumes half of its uplink bandwidth with UDP traffic, and thus its uplink bandwidth available for the conference reduces from 384 kbps to 192 kbps.

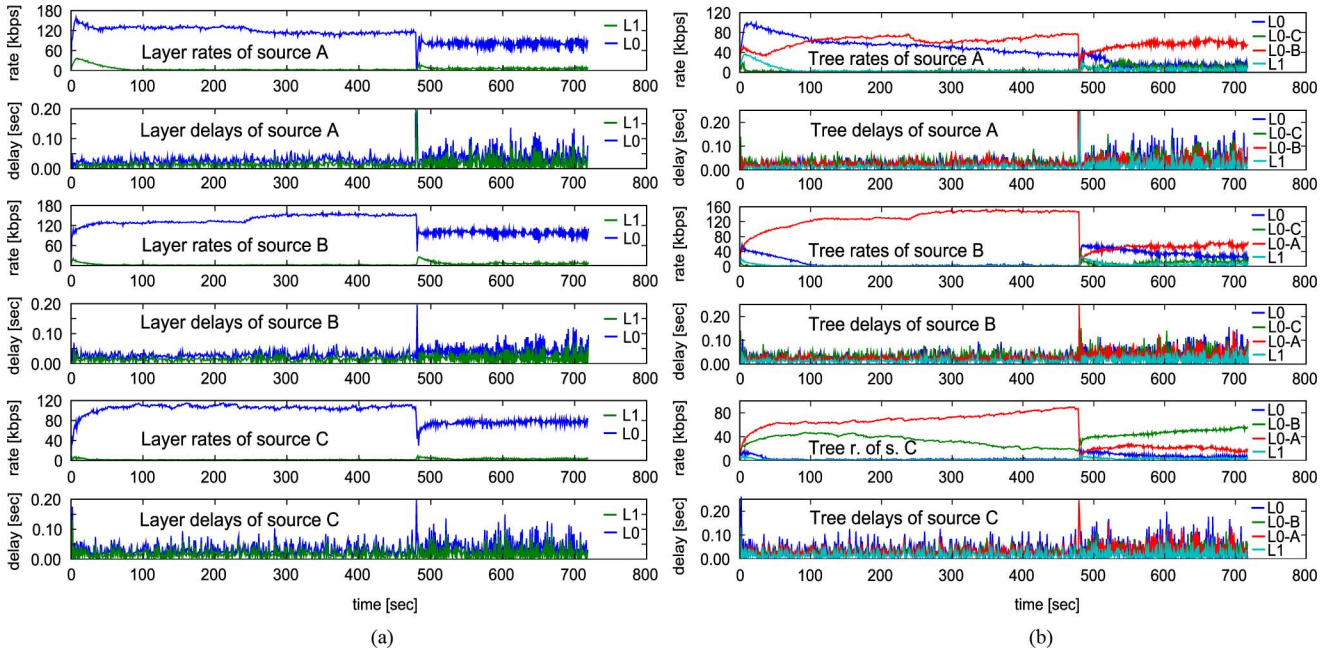
The experimental results are shown in Figs. 7 and 8(a). Fig. 7 shows the layer and individual tree rates, as well as the average and aggregate queuing delays of the trees. The aggregate queuing delay of a multicast tree is the sum of queuing delays along all branches of the tree and the average tree delay refers to an average of queuing delays of all trees used to deliver data for this layer. Fig. 8(a) shows the utilities of individual peers and the aggregate utility achieved by our system.

As seen in Fig. 7(b), the low-bandwidth peer C does not utilize its depth-1 tree, because it requires twice as much C's scarce bandwidth compared to sending content through high-bandwidth peers A or B. Moreover, for peers A and B, rates of the trees labeled by $L0 - C$ are close to zero. This indicates peers A and B do not use the low-bandwidth peer C to forward their video, allowing C to use its entire uplink bandwidth to distribute its own video.

At the 240th second, peer B's utility coefficient β_B increases. Seen from the increase in peer B's video rate in Fig. 7(a), our system reacts to this utility change by allocating more peer A's bandwidth to deliver B's video, thus optimizing the overall system-wise quality of experience. Peer A is chosen to be the victim because its utility coefficient is the same as peer C but it has more uplink bandwidth to help. The system's behavior makes intuitive sense.

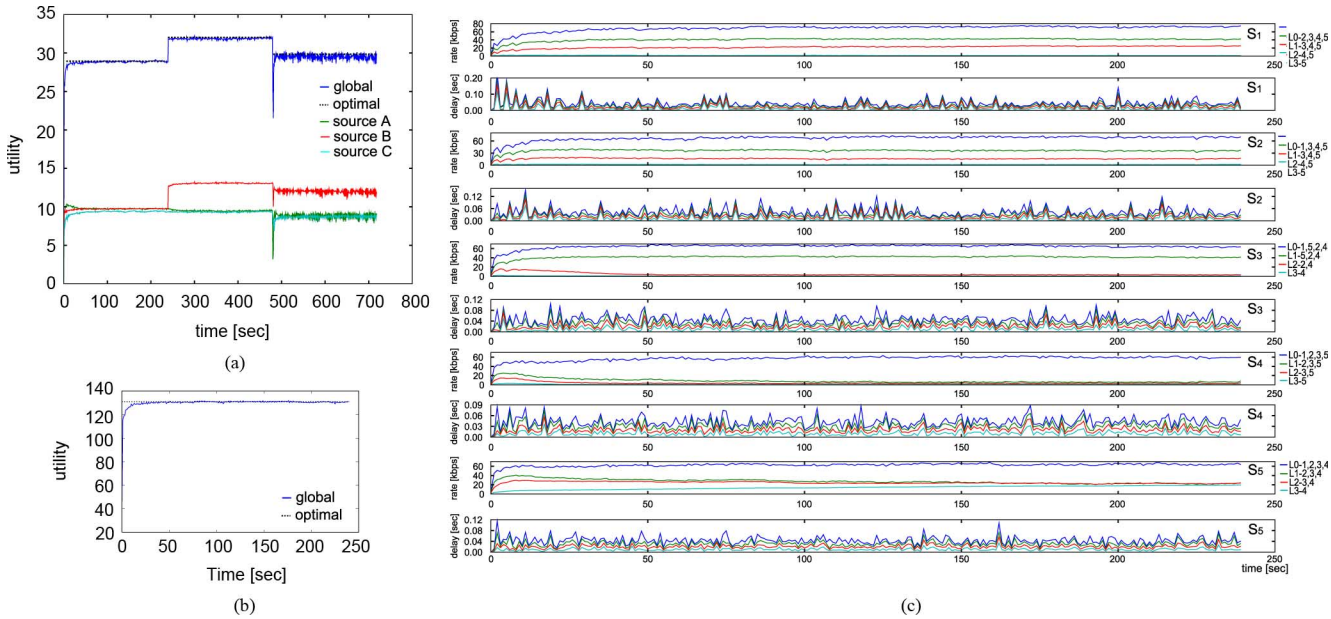
The cross traffic initiated at peer A at the 480th second causes an immediate drop in layer rates for all peers because peer A now has less bandwidth to forward their videos. Consequently, the queuing delay of peer A's uplink increases dramatically. The system quickly adapts to this change, and both tree rates and aggregate utility converge quickly to new optimal values.

All above observations highlight how the conferencing peers cooperate to maximize their overall video qualities in our



(a) (b)

Fig. 7. Experimental results for Scenario 1. (a) Layer rates ($L0$—base layer, $L1$—first enhancement layer) of sources A, B, and C, respectively, with the average tree queuing delays. (b) Tree rates for multicast trees of sources A, B, and C, respectively, with the aggregated tree queuing delays. Legends show the tree layer and also the intermediate node for depth-2 type trees.



(a) (b) (c)

Fig. 8. (a) Aggregate utility achieved by the system in Scenario 1 and the utilities per source. (b) Aggregate utility achieved by the system in Scenario 2. The optimal utility values are depicted by dotted lines in (a) and (b). (c) Layer rates ($L0$—base layer, Li − ith enhancement layer) of sources S_i, $1 \leq i \leq 5$, and the average tree queuing delays, in Scenario 2. Legend shows layers and the corresponding node indices of peers receiving the layers.

system, in the presence of network condition and conference characteristic changes.

We also observe in Fig. 7(b) that rates of the trees for enhancement layer videos are close to zero which is expected according to our established result for the receiver independent utility case (Theorem 3 and Corollary 1). Intuitively, this is because all receivers have the same utility, and optimally they should receive the source's video at the same rate, which is achieved by using only the trees for base layer video.

We can also see that even though the rates for individual multicast trees vary [Fig. 7(b)], the total layer rates converge quickly to the optimal solution [Fig. 8(a)] and stay relatively stable [Fig. 7(a)].

In order to verify the optimality of our distributed algorithm, we run Mosek [34] to solve Problem 2 under the settings of this experiment. The optimal tree rates allocation generated by Mosek confirms the optimality of our algorithm.

Our system takes 62 ms on average to deliver one packet from a sender to a receiver. If we distributed the videos in a simulcast way, it would be only about 20 ms but the peers would receive the videos at much lower quality, specifically for the peers with low uplink bandwidth. For instance, our system delivers peer

TABLE II
RECEIVER UTILITY COEFFICIENTS IN A FIVE-PARTY CONFERENCE IN SCENARIO 2

receiver r	S_1	S_2	S_3	S_4	S_5
$\beta_r^{S_1}$	-	0.5	0.75	1	1
$\beta_r^{S_2}$	0.5	-	0.75	0.875	0.875
$\beta_r^{S_3}$	0.5	1	-	1	0.5
$\beta_r^{S_4}$	0.5	0.5	0.5	-	0.5
$\beta_r^{S_5}$	0.5	0.65	0.8	1	-

C's video at rate 115 kbps, much higher than 64 kbps if simulcast approach has been used.

The optimization problem considered in this paper maximizes the utility based on received video bitrates and does not take network latency (propagation delays, bottleneck links in the middle of the network, and the associated variability in background Internet traffic) between any two peers directly into account, and we are thus neither minimizing nor explicitly bounding the total end-to-end packet delivery delay (compared to, for example, [20]). In our approach, packets travel through at most one intermediate node, and therefore, it may lead to lower average end-to-end delivery time compared to the server-assisted approach, such as [35], where packets always need to go through an intermediate server. On the other hand, in our approach, packet jitter can be higher due to multi-tree delivery. All packets required to decode a block of a video stream have to arrive in order to make any of the packets useful, and thus, the effective end-to-end delivery time, i.e., decoding delay, depends on the currently slowest route used to deliver data. Delay variability of individual packets is not as important as the time required for the last missing packet to arrive to be able to decode and play the stream. Our queueing delay-based approach automatically prefers routes which are not adding any extra delay and therefore tries to avoid slower routes with jitter in practice. Furthermore, by avoiding congestion via detecting increases in queuing delays, using network routes with high packet loss (due to congestion) is also minimized. We observe that the queuing delay introduced in the system by this approach is within the acceptable range for a smooth conferencing experience throughout the experiments in a controlled environment. Low end-to-end packet delivery delay is a requirement for high-quality video conferencing and a complete set of experiments in an uncontrolled environment to confirm the conferencing experience is left for future work.

B. Scenario 2: The Case of Diverse Peer Demands

With topology and peer uplink bandwidth shown in Fig. 5(b), we study a five-party conferencing scenario where propagation delay between peers are 20 ms. We choose source utility coefficients $\beta_{S_i}, (i = 1, \ldots, 5)$, to be the same, and set receiver's utility coefficients for sources S_1 to S_5 to values shown in Table II, representing highly diverse peer demands.

Under this setting, each peer needs to construct four video layers to meet the diverse peer demands. Each peer orders its receivers according to their receiver utility coefficients, forms layer session groups as described in Section II-B, and distributes its layered video to these session groups by using the depth-1 type and depth-2 type trees constructed by Algorithm 2. An example of peer S_1 distributing its four layers of video by using 13 trees is shown in Fig. 6(b).

We run the conference system for 250 seconds, and study the system performance in the presence of diverse receiver utility coefficients. Fig. 8(b) and (c) shows aggregate utility, layer rates, and average tree queuing delays. To satisfy the diverse needs, each peer uses more trees to deliver its video and forward others' videos. Thus, we have many more trees competing for uplink bandwidth than in Scenario 1, and the tree rates dynamics are expected to be more complex. Nevertheless, we can see from Fig. 8(b) and (c) that both the layer rates and aggregate utility still converge nicely and the achieved system utility is almost the same as the theoretically optimal one (computed by Mosek optimization package [34]). This shows that our system is capable of achieving good performance even under the complex conference setting studied in this scenario.

VI. CONCLUSION

We have presented a novel framework for multi-rate multi-source multicast that maximizes the aggregate utility of a small-scale P2P system. We show that by routing along a quadratic number of multicast trees per source, we can achieve the same rate region as that obtained through (intra-session) network coding in a P2P topology. We have developed Primal and Primal-dual distributed algorithms to maximize the aggregate utility and proved their global convergence. The developed algorithms are easy to implement in a P2P overlay over the current Internet infrastructure. Experimental results prove the usefulness of the proposed approach for multi-rate multiparty video conferencing applications where it maximizes the quality of experience for all participating peers, as predicted by our theoretical analysis. We demonstrate quick convergence to the optimal utility and automatic re-optimization when network conditions or conference characteristics change.

REFERENCES

[1] M. Ponec, S. Sengupta, M. Chen, J. Li, and P. A. Chou, "Multi-rate peer-to-peer video conferencing: A distributed approach using scalable coding," in *Proc. 2009 IEEE Int. Conf. Multimedia & Expo (ICME)*, New York, Jul. 2, 2009.

[2] M. Chen, M. Ponec, S. Sengupta, J. Li, and P. A. Chou, "Utility maximization in peer-to-peer systems," in *Proc. ACM SIGMETRICS*, Jun. 2008.

[3] V. Padman and N. Memon, "Design of a virtual laboratory for information assurance education and research," in *Proc. IEEE Workshop Information Assurance and Security*, West Point, NY, Jun. 2002.

[4] K. Kar, S. Sarkar, and L. Tassiulas, "Optimization based rate control for multirate multicast sessions," in *Proc. IEEE INFOCOM*, Anchorage, AK, Apr. 2001.

[5] S. Deb and R. Srikant, "Congestion control for fair resource allocation in networks with multicast flows," *IEEE Trans. Autom. Control*, vol. 49, no. 4, pp. 274–285, Apr. 2004.

[6] K. Kar and L. Tassiulas, "Layered multicast rate control based on Lagrangian relaxation and dynamic programming," *IEEE J. Select. Areas Commun.*, vol. 24, no. 8, pp. 1464–1474, Aug. 2006.

[7] L. Chen, T. Ho, S. H. Low, M. Chiang, and J. C. Doyle, "Optimization based rate control for multi-cast with network coding," in *Proc. IEEE INFOCOM*, Anchorage, AK, May 2007.

[8] Y. Wu and S.-Y. Kung, "Distributed utility maximization for network coding based multicasting: A shortest path approach," *IEEE J. Select. Areas Commun.*, vol. 24, no. 8, pp. 1475–1488, Aug. 2006.

[9] D. S. Lun, N. Ratnakar, M. Médard, R. Koetter, D. R. Karger, T. Ho, E. Ahmed, and F. Zhao, "Minimum-cost multicast over coded packet networks," *IEEE Trans. Inf. Theory*, vol. 52, no. 6, pp. 2608–2623, Jun. 2006.

[10] D. M. Chiu, R. Yeung, J. Huang, and B. Fan, "Can network coding help in P2P networks?," in *Proc. IEEE 2nd Workshop Network Coding (NetCod)*, Boston, MA, Apr. 2006, p. 2.

[11] J. Mo and J. Walrand, "Fair end-to-end window-based congestion control," *IEEE/ACM Trans. Network.*, no. 5, pp. 556–567, Oct. 2001.

[12] F. P. Kelly, A. Maulloo, and D. Tan, "Rate control for communication networks: shadow prices, proportional fairness, and stability," *J. Oper. Res. Soc.*, pp. 237–252, 1998.

[13] L. S. Brakmo and L. L. Peterson, "TCP Vegas: End-to-end congestion avoidance on a global internet," *IEEE J. Select. Areas Commun.*, vol. 13, no. 8, pp. 1465–1480, Oct. 1995.

[14] R. La, J. Walrand, and V. Anantharam, Issues in TCP Vegas, Univ. California, Berkeley, 1998.

[15] M. Welzl and D. Ros, A Survey of Lower-Than-Best-Effort Transport Protocols, Internet-Draft, 2011.

[16] A. Norberg, UTorrent Transport Protocol Draft. [Online]. Available: http://www.bittorrent.org/beps/bep_0029.html.

[17] S. Shalunov, G. Hazel, J. Iyengar, and M. Kuehlewind, Low Extra Delay Background Transport (LEDBAT), Internet-Draft, 2011.

[18] C. Liang, M. Zhao, and Y. Liu, "Optimal bandwidth sharing in multi-swarm multi-party P2P video conferencing systems," *IEEE/ACM Trans. Netw.*, accepted for publication.

[19] I. Akkus, O. Ozkasap, and M. Civanlar, "Peer-to-peer multipoint video conferencing with layered video," *J. Netw. Comput. Appl.*, vol. 34, no. 1, pp. 137–150, 2011.

[20] X. Chen, M. Chen, B. Li, Y. Zhao, Y. Wu, and J. Li, "Celerity: Towards low delay multiparty conferencing over arbitrary network topologies," in *Proc. 21th Int. Workshop Network and Operating Systems Support for Digital Audio and Video (ACM NOSSDAV 2011)*, Vancouver, BC, Canada, Jun. 2011.

[21] V. K. Goyal, "Multiple description coding: Compression meets the network," *IEEE Signal Process. Mag.*, vol. 18, no. 5, pp. 74–93, Sep. 2001.

[22] R. Puri and K. Ramchandran, "Multiple description source coding through forward error correction codes," in *Proc. IEEE Asilomar Conf. Signals, Systems, and Computers*, Oct. 1999.

[23] H. Schwarz, D. Marpe, and T. Wiegand, "Overview of the scalable H.264/MPEG4-AVC extension," in *Proc. IEEE Int. Conf. Image Processing (ICIP'06)*, Oct. 2006.

[24] H. Schwarz, D. Marpe, and T. Wiegand, "Overview of the scalable video coding extension of the H.264/AVC standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 9, pp. 1103–1120, Sep. 2007.

[25] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung, "Network information flow," *IEEE Trans. Inf. Theory*, vol. 46, no. 4, pp. 1204–1216, Jul. 2000.

[26] X. Yan, R. W. Yeung, and Z. Zhang, "The capacity region for multi-source multi-sink network coding," in *Proc. 2007 IEEE Int. Symp. Information Theory (ISIT2007)*, Nice, France, Jun. 2007.

[27] J. Li, P. A. Chou, and C. Zhang, "Mutualcast: An efficient mechanism for content distribution in a P2P network," in *Proc. ACM SIGCOMM Asia Workshop*, 2005, p. 2.

[28] S. Floyd, "TCP and Explicit Congestion Notification," *ACM Comput. Commun. Rev.*, pp. 10–23, Oct. 1994.

[29] M. Chen, "A general framework for flow control in wireless networks," Ph.D. dissertation, Univ. California, Berkeley, 2006.

[30] S. H. Low, L. Peterson, and L. Wang, "Understanding Vegas: A duality model," *J. ACM*, vol. 49, no. 2, pp. 207–235, Mar. 2002.

[31] S. Liu, T. Basar, and R. Srikant, "Exponential-RED: A Stabilizing AQM Scheme for Low- and High-Speed TCP Protocols," *IEEE/ACM Trans. Netw.*, vol. 5, Oct. 2005.

[32] T. Voice, "Stability of congestion control algorithms with multi-path routing and linear stochastic modelling of congestion control," Ph.D. dissertation, Univ. Cambridge, Cambridge, U.K., 2006.

[33] M. Jain and C. Dovrolis, "End-to-end available bandwidth: Measurement methodology, dynamics, and relation with tcp throughput," *IEEE/ACM Trans. Netw.*, vol. 11, pp. 537–549, Aug. 2003.

[34] MOSEK ApS,MOSEK Optimization Software. [Online]. Available: http://www.mosek.com/products_mosek.html.

[35] E. Alexandros, C. M. Reha, and S. Ofer, "Multipoint videoconferencing with scalable video coding," *J. Zhejiang Univ.—Sci. A*, vol. 7, pp. 696–705, Apr. 2006.

Miroslav Ponec received the M.S. degree from Czech Technical University, Prague, Czech Repulic, in 2005 and the Ph.D. degree from the Computer and Information Science Department at Polytechnic University, Brooklyn, NY, in 2008.

He is currently at Akamai Technologies GmbH, Unterfoehring, Germany.

Sudipta Sengupta received the B.Tech. degree in computer science from Indian Institute of Technology (IIT), Kanpur, India, and the M.S. degree in electrical engineering and computer science and the Ph.D. degree from the Massachusetts Institute of Technology (MIT), Cambridge.

He is currently at Microsoft Research, Redmond, WA.

Minghua Chen received the B.Eng. and M.S. degrees from the Department of Electronics Engineering at Tsinghua University, Beijing, China, in 1999 and 2001, respectively, and the Ph.D. degree from the Department of Electrical Engineering and Computer Sciences at the University of California at Berkeley in 2006.

He is currently professor at the Chinese University of Hong Kong, Shatin, Hong Kong.

Jin Li received the B.S., M.S., and Ph.D. degrees from the Electronic Engineering Department, Tsinghua University, Beijing, China, all with honors.

He is currently with Microsoft Research, Redmond, WA.

Philip A. Chou received the B.S.E. degree from Princeton University, Princeton, NJ, in 1980, and the M.S. degree from the University of California, Berkeley, in 1983, both in electrical engineering and computer science, and the Ph.D. degree in electrical engineering from Stanford University, Stanford, CA, in 1988.

He is currently at Microsoft Research, Redmond, WA.