MULTI-RATE PEER-TO-PEER VIDEO CONFERENCING: A DISTRIBUTED APPROACH USING SCALABLE CODING

Miroslav Ponec[*] *Sudipta Sengupta*[†] *Minghua Chen*[‡] *Jin Li*[†] *Philip A. Chou*[†]

[*]Akamai Technologies and Polytechnic Institute of NYU. mponec@akamai.com
[†]Microsoft Research, Redmond, WA. {sudipta,jinl,pachou}@microsoft.com
[‡]The Chinese University of Hong Kong. minghua@ie.cuhk.edu.hk

ABSTRACT

We consider multi-rate peer-to-peer multi-party 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 rate. We study and address the unique challenges introduced by 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 offers low end-to-end delay, low complexity and high throughput, along with automatic adaptation to network conditions and user preferences.

1. INTRODUCTION AND MOTIVATION

Providing Quality-of-Service (QoS) in P2P multi-party 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 the Peak-Signal-to-Noise-Ratio (PSNR) of the decoded video.

There are several existing solutions for conducting P2P multi-party conferencing. The client-server approach ensures that the entire upload bandwidth of each peer can be used for the delivery of just that peer's audio/video session. However, the central machine may suffer a heavy burden of CPU and network bandwidth from serving many conferencing sessions simultaneously. In the ad hoc simulcast approach, each peer splits its upload bandwidth equally for all other peers and sends a copy of its video to each peer separately. Though simple to implement, this approach suffers from poor quality of service as the peer with the minimum upload bandwidth is forced to use a low coding rate that degrades the overall experience of the other peers. In recent work [1], overlay routing and allocation of source rates in a P2P multi-party 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. For instance, by using a scalable video codec, sources can generate one video stream that can be decoded at different rates. As a result, receivers with larger screens 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 multi-party 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 for each source in the overlay network, whose number is *quadratic* in the number of nodes.
- **Multi-tree Based Formulation and Distributed Algorithms:** We give a new multi-tree based formulation for P2P multi-rate multicast utility maximization, 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 queu-

ing 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 multi-party conferencing system using the delay-based Primal-dual algorithm, and evaluated its performance over the Virtual Lab testbed [2]. The results show that the system can achieve the optimal utility as predicted by theoretical analysis. The strict end-to-end packet delivery delay requirements for conferencing is also satisfied.

The proofs of all theorems in this paper can be found in [3].

1.1. Related Work

Utility maximization based rate control for multicast routing is a well-studied problem, 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. Almost all 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 multi-party conferencing [1]. 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.

2. PROBLEM STATEMENT

The key notations used in this paper are listed in Table 1. 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 $\boldsymbol{G^s} = \{G_\ell^s, 0 \le \ell \le |R_s| - 1\}$.

2.1. 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.

Table 1. 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 leyer
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

There are two common approaches for sources to provide multi-rate streams, namely *Multiple Description Coding* (MDC) and *Layered Coding*. MDC 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), 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 Scalable Video Coding (SVC) in our approach where both the number of layers each user receives and the layer rates together provide the video quality scalability.

2.2. Layer Assignment

Suppose for a given source s, the receiver rates are ordered as $x_{i_1}^s \le x_{i_2}^s \le \ldots \le x_{i_{|R_s|}}^s$. 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 $(x^s_{i_2} - x^s_{i_1})$ 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 $(x^s_{i_{\ell+1}} - x^s_{i_\ell})$ and is multicasted from s to all receivers in $\{i_{\ell+1}, i_{\ell+2}, \ldots, i_{|R_s|}\}$.

2.3. 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 [4]. Depending on whether packets from different sessions are mixed or not, 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; however, computing such mixing and coding is still a largely open problem.

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

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

$$\sum_{e \in E^+(i)} y_e^{s\ell r} \quad - \quad \sum_{e \in E^-(i)} y_e^{s\ell r} = \begin{cases} +z^s_\ell & \text{if } i = s \\ -z^s_\ell & \text{if } i = r \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

$$\forall\, i \in N, r \in \{i_{\ell+1}, \ldots, i_{|R_s|}\}, 0 \le \ell \le |R_s| - 1, s \in S$$

$$\sum_{s \in S} \sum_{\ell=0}^{|R_s|-1} \max_{r \in R_s}(y_e^{s\ell r}) \quad \le \quad C_e \;\; \forall\, e \in E \tag{2}$$

$$z^s_0 \quad = \quad \min_{r \in R_s}(x^s_r) \;\; \forall\, s \in S$$

$$z^s_\ell \quad = \quad \min^{\ell+1}_{r \in R_s}(x^s_r) - \min^{\ell}_{r \in R_s}(x^s_r)$$

$$\forall\, 1 \le \ell \le |R_s| - 1, s \in S$$

where $E^-(i)$ denotes the links going into node i and $E^+(i)$ links leaving node i, and $\min^\ell$ to denote the ℓ-th 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 ℓ-th 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

Problem 1 (Multi-rate Multicast Utility Maximization)

$$\max_{\boldsymbol{x}} \quad \textstyle\sum_{s \in S} \sum_{r \in R_s} U^s_r(x^s_r), \;\; s.t. \quad \boldsymbol{x} \in \mathcal{B}. \tag{3}$$

2.4. 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 [5][1], *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 [5] 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: (i) One depth-1 tree rooted at s and reaching all receivers in R_s, (ii) $|R_s|$ depth-2 trees, each rooted at s and reaching all other receivers in R_s via different $r \in R_s$, and (iii) $|H|$ 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 [1].

We now extend the above result to the *multi-source multi-rate* scenario, for which we need the depth-1 and depth-2 trees to be more flexible 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 are shown in Fig. 1.

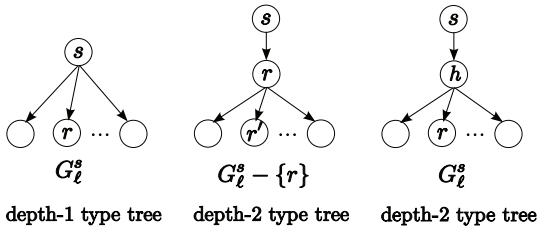

Fig. 1. Depth-1 type and depth-2 type multicast trees. Here G^s_ℓ serves as an example subset of R_s.

Suppose we know the ordering of receiver rates $x^s_r, r \in R_s$ 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 $\prod_{s \in S} |R_s|!$ We use π^s_i to denote the i^{th} 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^s_r 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.

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 1.*

Algorithm 1 : Layer Trees Construction.

```
// Input: Session group G^s of source s
// Function: Construct depth-1 type and depth-2 type trees to deliver s's layered video
for ℓ from 0 to |R_s| − 1 do
    Construct a depth-1 type tree reaching all receivers in G^s_ℓ from s
    for r ∈ G^s_ℓ do
        Construct one depth-2 type tree reaching r from s, and then to the rest of receivers in G^s_ℓ − {r}
    end for
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.*

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

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

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 [1] 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.*

2.5. Tree-based Formulation For P2P Multi-rate Multicast Utility Maximization

For a tree m with rate ξ_m. Receiver nodes on a tree receive the same content at the same rate. With 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

$$\lambda_e = \sum_{s \in S} \sum_{m: m \in s, e \in m} b^m_e \xi_m, \quad \forall e \in E, \qquad (5)$$

where b^m_e 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^m_e. Based on Theorem 2, we reformulate Problem 1 as follows:

Problem 2 (Tree-based Multi-rate Multicast Utility Maximization)

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

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.

3. DISTRIBUTED ALGORITHMS

3.1. 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:

$$\max_{\boldsymbol{\xi}} \sum_{s \in S} \sum_{r \in R_s} U^s_r \left(\sum_{m: m \in s, r \in m} \xi_m \right) - \sum_{e \in E} \int_0^{\lambda_e} q_e(w)\, dw, \qquad (7)$$

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 [6]. The strict concavity of $U^s_r(\cdot)$ 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 $q_e(w) = \frac{(w - C_e)^+}{w}$ for link e. In terms of ECN marking [7], it represents the packet marking probability. We consider the following Primal algorithm: $\forall s \in S, \forall m \in s$,

$$\dot{\xi}_m = f_m(\xi_m) \left(\sum_{r \in m} U'^s_r \left(\sum_{m: m \in s, r \in m} \xi_m \right) - \sum_{e \in m} b^m_e q_e(\lambda_e) \right), \qquad (8)$$

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. [6]. Moreover, it is also possible to show that the convergence is actually semi-globally exponentially fast. We skip the proofs due to space limitation.

3.2. A Queuing Delay Based Primal-dual Algorithm

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

$$L(\boldsymbol{\xi}, \boldsymbol{p}) = \sum_{s\in S}\sum_{r\in R_s} U_r^s\left(\sum_{m:m\in s, r\in m}\xi_m\right) - \sum_{e\in E} p_e(\lambda_e - C_e), \quad (9)$$

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: $\forall s \in S, \forall m \in s, \forall e \in E$,

$$p_e \geq 0, \quad \lambda_e \leq C_e, \quad p_e(\lambda_e - C_e) = 0, \quad (10)$$

$$\sum_{r\in m} U_r'^s\left(\sum_{m:m\in s, r\in m}\xi_m\right) - \sum_{e\in m} b_e^m p_e = 0. \quad (11)$$

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 $e \in E$,

$$\dot{\xi}_m = k_m\left(\sum_{r\in m} U_r'^s\left(\sum_{m:m\in s, r\in m}\xi_m\right) - \sum_{e\in m} b_e^m p_e\right)_{\xi_m}^+, \quad (12)$$

$$\dot{p}_e = \frac{1}{C_e}(\lambda_e - C_e)_{p_e}^+, \quad (13)$$

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 (13) can be interpreted as queuing delay [8] on uplink e.

Under multi-tree/multi-path delivery setting, it is shown that the queuing delay p following (13) can oscillate indefinitely and may never converge [9, Section 2.5]. In our previous work in [1], we give a sufficient condition for the Primal-dual system in (12)-(13) 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* multi-party 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 [1] is satisfied.

Theorem 3 *For P2P multi-rate multi-party conferencing scenario, all trajectories of the system in (12)-(13) converge to one of its equilibria globally asymptotically, if k_m are the same for all the trees $m \in s$.*

The Primal-dual algorithm described in (12)-(13) 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 4.

4. PRACTICAL IMPLEMENTATION

We have implemented the queuing delay based distributed algorithm (12)-(13) in a prototype of a P2P multi-rate multi-party video conferencing system. 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 (12), based on the measured queuing delays it collected from other peers.

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

4.1. Utility Modeling and Layer Assignment

Peak signal-to-noise ratio (PSNR) is the de facto standard metric in video processing to provide objective quality evaluation between the original frame and the compressed one. We found empirically 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. (Further details provided in [3].)

In our implementation, when a peer r subscribes to a video stream of source s it submits a *receiver utility coefficient*, de-

noted 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 strictly concave.

Source s also sorts all receivers according to their receiver utility coefficients. Assuming that the receiver rates also follow this order, the source s determines the number of layers to construct, assigns layers to receivers as described in Section 2.2, and builds the set of trees to distribute these layers of video according to Algorithm 1.

4.2. 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.

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 (12).

5. EXPERIMENTAL RESULTS

We use a set of virtual machines in a Virtual Lab infrastructure [2] to conduct experiments in Scenarios 1 and 2 to evaluate the performance of our multi-party conferencing prototype described in Section 4.

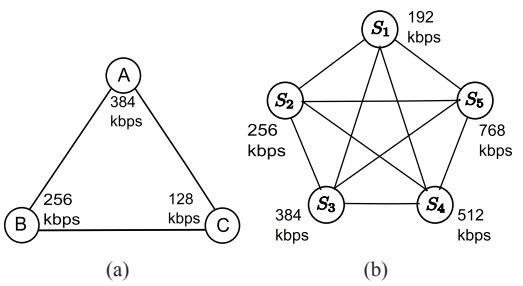

Fig. 2. (a) Topology of Scenario 1 and peer uplink bandwidth setting. (b) Topology of Scenario 2 and peer uplink bandwidth setting.

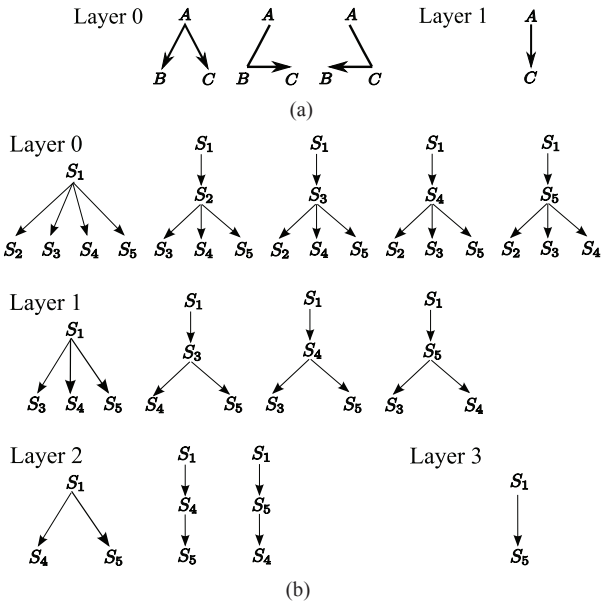

Fig. 3. (a) Multicast trees delivering data of video layers of source A in a 3-party conference in Scenario 1. (b) Multicast trees delivering data of video layers of source S_1 in a 5-party conference in Scenario 2.

5.1. 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. 2(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 4.1. 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 are sent along a set of depth-1 and depth-2 trees which are constructed according to the procedure in Section 2.3. For instance, as shown in shown in Fig. 3(a), peer A uses three trees to send its base layer video, and use 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 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.

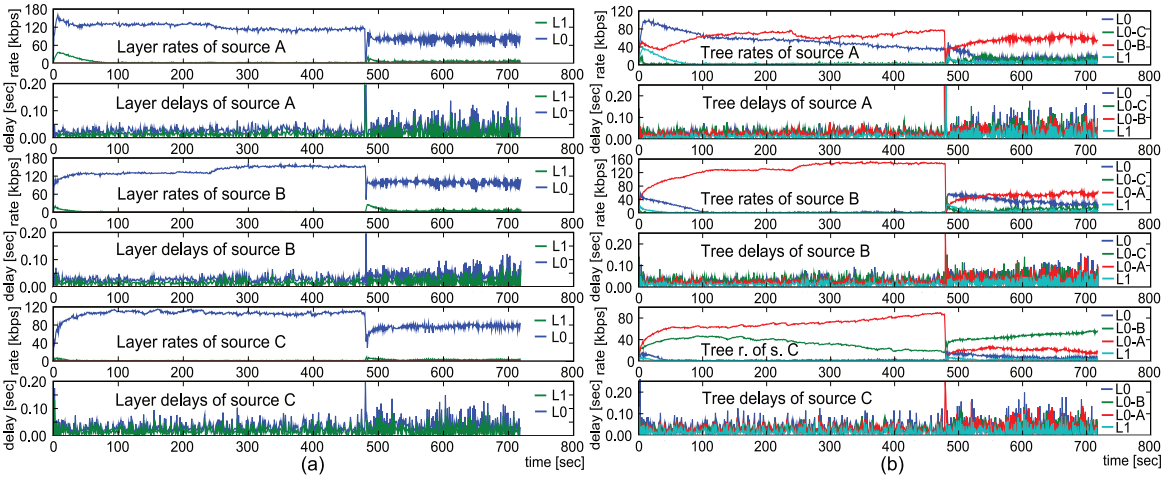

Fig. 4. Experimental results for Scenario 1: (a) Layer rates ($L0$ - base layer, $L1$ - first enhancement layer) of sources A, B, and C, respectively, with the average tree queuing delays. (b) Tree rates for multicast trees of sour ces A, B and C, respectively, with the aggregated tree queuing delays. Legends show the tree layer and also the intermediate node for depth-2 type trees.

The experimental results are shown in Fig. 4 and Fig. 5(a). Fig. 4 shows the layer and individual tree rates, as well as the average and aggregate queuing delays of the trees. Fig. 5(a) shows the utilities of individual peers and the aggregate utility achieved by our system.

As seen in Fig. 4(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 240th second, peer B's utility coefficient β_B increases. Seen from the increase in peer B's video rate in Fig. 4(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 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 system, in the presence of network condition and conference characteristic changes.

We also observe in Fig. 4(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 (omitted here but available in [3]). 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. 4(b)), the total layer rates converge quickly to the optimal solution (Fig. 5(a)) and stay relatively stable (Fig. 4(a)).

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

5.2. Scenario 2: The case of diverse peer demands

With topology and peer uplink bandwidth shown in Fig. 2(b), we study a 5-party conferencing scenario where propagation delay between peers are 20 ms and peers have highly diverse demands. Serving this purpose, we choose source utility coefficients β_{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 between 0.5 and 1 (specified in [3]).

Under this setting, each peer needs to construct 4 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 2.2, and distributes its layered video to these session groups by using the depth-1 type and depth-2 type trees constructed by Algorithm 1. An example of peer S_1 distributing its 4 layers of video by using 13 trees are shown in Fig. 3(b).

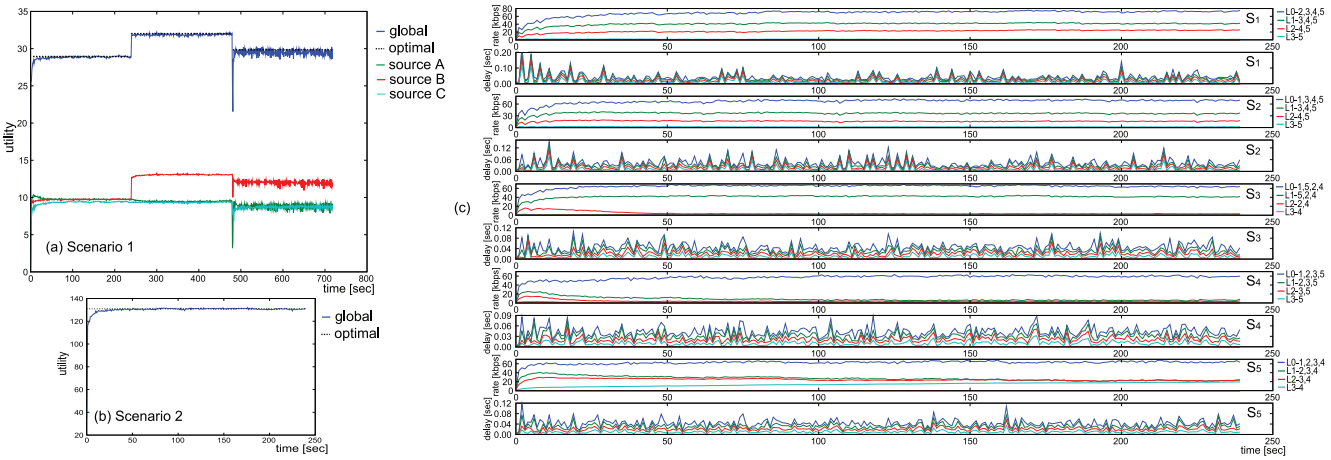

Fig. 5. (a) The aggregate utility achieved by the system in Scenario 1 and the utilities per source. (b) The aggregate utility achieved by the system in Scenario 2. The optimal utility values are depicted by dotted lines in (a) and (b). (c) Layer rates ($L0$ - base layer, Li - i-th enhancement layer) 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.

We run the conference system for 250 seconds, and study the system performance in the presence of diverse peer demands. Fig.s 5(b,c) show aggregate utility, layer rates, and average tree queuing delays. To satisfy the diverse peer demands, 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 Figs. 5(b,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). This shows that our system is capable of achieving good performance even under the complex conference setting studied in this scenario.

6. CONCLUSION

We have presented a novel framework for multi-rate multi-source multicast that maximizes the aggregate utility of a P2P system. The nature of P2P topologies allows us to solve the difficulties arising in the general network case. 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. We have developed Primal and Primal-dual distributed algorithms to maximize the aggregate utility and proved their global convergence. The developed algorithms are practical and easy to implement in a P2P overlay over the current Internet. Experimental results prove the usefulness of the proposed approach for multi-rate multi-party 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.

7. REFERENCES

[1] Minghua Chen, Miroslav Ponec, Sudipta Sengupta, Jin Li, and Philip A. Chou, "Utility maximization in peer-to-peer systems," *ACM SIGMETRICS 2008*, June 2008.

[2] Vikram Padman and Nasir Memon, "Design of a virtual laboratory for information assurance education and research," in *Proceedings of the 2002 IEEE Workshop on Information Assurance and Security*, West Point, NY, June 2002.

[3] Miroslav Ponec, Sudipta Sengupta, Minghua Chen, Jin Li, and Philip A. Chou, "Multi-rate peer-to-peer video conferencing: A distributed approach using scalable coding," Microsoft Technical Report, August 2008.

[4] Rudolf Ahlswede, Ning Cai, Shuo-Yen Robert Li, and Raymond W. Yeung, "Network information flow," , no. 4, pp. 1204–1216, July 2000.

[5] Jin Li, Philip A. Chou, and Cha Zhang, "Mutualcast: an efficient mechanism for content distribution in a p2p network," in *Proceedings of ACM SIGCOMM Asia Workshop*, 2005.

[6] F. P. Kelly, A. Maulloo, and D. Tan, "Rate control for communication networks: shadow prices, proportional fairness, and stability," *Journal of the Operationl Research Society*, pp. 237–252, 1998.

[7] S. Floyd, "TCP and explicit congestion notification," *ACM Computer Communication Review*, pp. 10–23, Oct. 1994.

[8] S. H. Low, Larry Peterson, and Limin Wang, "Understanding Vegas: A duality model," *Journal of ACM*, vol. 49, no. 2, pp. 207–235, Mar. 2002.

[9] Thomas Voice, *Stability of Congestion Control Algorithms with Multi-Path Routing and Linear Stochastic Modelling of Congestion Control*, Ph.D. thesis, University of Cambridge, Cambridge, UK, May 2006.