Rate Distortion Optimization for Mesh-based P2P Video Streaming

Tareq Hossain, Yi Cui, and Yuan Xue
Vanderbilt Advanced NETwork Systems Laboratory
Department of Electrical Engineering and Computer Science
Vanderbilt University, Nashville, TN 37212
Email: {tareq.hossain, yi.cui, yuan.xue}@vanderbilt.edu

***Abstract*—This paper addresses the problem of optimal rate allocation for video streaming in a multi-path peer-to-peer mesh network. We present a distributed rate allocation algorithm that minimizes the total rate distortion among receiving peers. The scheme assumes that video streams can be transcoded/re-quantized at intermediate peers. We deploy a double pricing solution that simultaneously incorporates both the network and the relay constraints. We compare it with a single pricing solution where the relay constraint is applied only after all the communicating peers have converged. Our simulation shows that the double pricing solution consistently achieves a smaller aggregate distortion for all peers in comparison to the single pricing solution and thus achieves higher video quality.**

I. Introduction

Peer-to-peer (P2P) is a powerful platform for a variety of multimedia streaming applications over the Internet such as video-on-demand, video conferencing, live broadcasting, etc. A P2P system is extremely cost-effective since it utilizes the resources (CPU cycles, storage space, and uplink bandwidth) of peer machines. Another reason for P2P's success is its instant deployability; it allows almost ubiquitous network coverage in the absence of CDN services and IP multicast. In the case of video streaming, content is delivered in a distributed environment where peers act as end hosts of an overlay mesh.

Existing approaches for P2P streaming can be divided into two classes: *tree-based* and *mesh-based*. The tree-based approach extends the idea of end-system multicast [1]. A mesh-based P2P streaming is derived from file swarming mechanisms (such as BitTorrent) where participating peers form a randomly connected mesh. Due to the multiple incoming connection for each peer, a multi-path mesh can fully utilize the network resources of its peers. Furthermore, peers experience a higher degree of stability [2] in a mesh-based approach compared to a tree-based approach. This results in higher quality for the delivered video.

In this paper, we present an optimal rate allocation algorithm for mesh-based P2P video streaming applications. Derived from Kelly's network optimization framework [3], [4], the algorithm seeks to minimize the aggregate rate distortion for all peers. Each peer adjusts its own streaming rate to reach the global optimum in our distributed algorithm. The following are our contributions.

The optimization problem forumulation is based on our previous work [5] on rate allocation optimization for overlay multicast tree. We take into account peer relaying - a constraint unique to a P2P distribution scenario in which a peer is both receiver and sender - and extend this to mesh-based P2P networks. Specifically, we propose a receiver-driven distibuted algorithm that minimizes rate distorion in a mesh-based P2P network. Our solution is combined with the transcoding video adaptation technique.

The rest of this paper is organized as follows. In Sec. II, we introduce the network model. In Sec. III, we present the formulation for minimizing aggregate rate distortion and propose a distributed solution for a multi-path mesh scenario. Sec. IV presents the mesh construction procedure, rate adaptation via transcoding, and simulation results. Finally, we discuss related work in Sec. V and conclude in Sec. VI.

II. Mesh Network Model

We consider a P2P network consisting of H end hosts, denoted as $\mathcal{H} = \{0, 1, 2, \cdots, H\}$. Host 0 acts as the server, and other end hosts are peers. A flow f can direct from a peer to another peer except from a peer to the server 0. We collect all these flows into the set $\mathcal{F} = \{1, 2, \cdots, F\}$. Each flow $f \in \mathcal{F}$ has a rate x_f. The rate vector is then defined as $\boldsymbol{x}$ = $(x_f, f \in \mathcal{F})$. For example, consider two peers: h_i and h_j. If a flow f_{ij} directs from h_i to h_j, then h_i is a parent of h_j and h_j is a child of h_i. This relationship is then denoted as $h_i \rightarrow h_j$. In the above example, the flow f directs to the peer h_j which we denote $h(f)$. Based on this, we also introduce $f(h)$ as the flow destined to peer h.

For a peer $h_i \in \mathcal{H}$, we define $\mathcal{S}(h_i) = \{f_{ij} \,|\, \forall j, h_i \rightarrow h_j\}$ to be the set of flows directed to its child peers. Similarly, we define $\mathcal{R}(h_i) = \{f_{ki} \,|\, \forall k, h_k \rightarrow h_i\}$ to be the set of flows directed to peer h_i from all its parents.

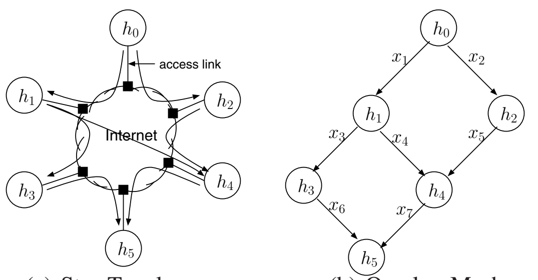
Each flow f takes place on the unicast path that connects two peers and encompasses a set of physical links on the Internet. We collect all physical links encompassed by all flows in $\mathcal{F}$ into a vector as $\mathcal{L} = \{1, 2, \cdots, L\}$. Therefore, $\mathcal{L}(f)$ is the set of physical links that flow f goes through. The bandwidth of each link is defined as c_l which is collected into a capacity vector defined as $\boldsymbol{c}$ = $(c_l, l \in \mathcal{L})$. Based on this definition, for each link l, we define its flow set

$\mathcal{F}(l) = \{f \in \mathcal{F} \mid l \in \mathcal{L}(f)\} = \{\mathcal{S} \cup \mathcal{R}\}$ as the set of flows that pass through it.

We now present two sets of constraints used to formulate our problem: *capacity constraint* and *relay constraint*. The capacity constraint states that for each link l, the total rate for all the flows in $\mathcal{F}(l)$ cannot exceed the link capacity c_l, i.e., $\sum_{f \in \mathcal{F}(l)} x_f \leq c_l$. Formally, let $\boldsymbol{A}$ be an $L \times F$ matrix, such that $A_{lf} = 1$ if flow f goes through the link l, otherwise $A_{lf} = 0$.

$$\boldsymbol{A} \cdot \boldsymbol{x} \leq \boldsymbol{c} \tag{1}$$

In this paper, our physical topology is based on the assumption [6] that that the bottleneck link of an end-to-end connection only happens at the uplink of the sending peer. As such, since the capacity constraint of other links never gets violated, they can be removed from the above inequality. This effectively reduces the link set $\mathcal{L}$ to only contain the uplinks of all peers and the server. As illustrated in Fig. 1(a), we give such a special topology the term *star topology*. Under this topology, the size of the link set equals the size of the end host set $\mathcal{H}$. The relay constraint states that the receiving rate of a peer cannot exceed the receiving rate of its parent. We illustrate this idea with the overlay mesh shown in Fig. 1(b). In this picture, the relay constraint states that the video quality received by h_3 and h_4 cannot be higher than the quality received by their parent h_1. Therefore, flow rate x_3 and x_4 cannot exceed flow rate x_1. In the case of peer h_4, it receives data from both peer h_1 and h_2. Consequently, the outgoing flow rate of h_4 can not exceed the total incoming flow rate. Formally, the relay constraint states that $x_7 - x_4 - x_5 \leq 0$.



(a) Star Topology (b) Overlay Mesh

Fig. 1. P2P Streaming Illustration

Since any peer can be the parent of any other peer, the total number of such parent-child pairs[1] is H^2. We formulate the relay constraint in a $H^2 \times F$ matrix $\boldsymbol{B}$ as follows:

$$B_{((h_k-1)H+h_i)\cdot f} = \begin{cases} -1 & \text{if } h_k = h(f) \text{ and } h_k \rightarrow h_i \\ 1 & \text{if } h_i = h(f) \text{ and } h_k \rightarrow h_i \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

[1]However, there are several special cases which forbid parent-child pairs. For example, the server h_0 cannot be the child of any peer, a peer cannot be the parent of itself and there can be no cylclic paths (such as path $h_3 \rightarrow h_5 \rightarrow h_4 \rightarrow h_3$) in the network. For simplicity purposes, we avoid such network formations. Nevertheless, the actual number of parent-child pair number remains in the order of H^2.

TABLE I
NETWORK MODEL & DISTRIBUTED ALGORITHM NOTATIONS

Notation	Definition
$h \in \mathcal{H} = \{1, 2, \cdots, H\}$	End host
$f \in \mathcal{F} = \{1, 2, \cdots, F\}$	Unicast flow in overlay multicast
$\boldsymbol{x} = (x_f, f \in \mathcal{F})$	Flow rate of $f \in \mathcal{F}$
$l \in \mathcal{L} = \{1, 2, \cdots, L\}$	Physical network link
$\boldsymbol{c} = (c_l, l \in \mathcal{L})$	Link capacity of $l \in \mathcal{L}$
$h_k \rightarrow h_i$	Host h_i is the child of host h_k
$\mathcal{L}(f) \subseteq \mathcal{L}$	Set of links that f goes through
$\mathcal{F}(l) \subseteq \mathcal{F}$	Set of flows that go through link l
$h(f)$	Peer that flow f directs to
$f(h)$	Flow that directs to peer h
$\mathcal{S}(h_i)$	Set of outgoing flows of host h_i
$\mathcal{R}(h_i)$	Set of incoming flows of host h_i
$\boldsymbol{A} = (A_{lf})_{L \times F}$	Link capacity Constraint Matrix
$\boldsymbol{B} = (A_{f'f})_{F \times F}$	Data constraint Matrix
$D_f(x_f)$	Rate distortion function
μ_l^α	Link price for link l
μ_f^β	Relay price for flow f
λ_f^α	Sum of link prices in $\mathcal{L}(f)$
λ_f^β	Aggregate relay prices

$\boldsymbol{B}$ is a sparse matrix, where row $((h_k - 1)H + h_i)$ will only be active if there is a flow from h_k to h_i. The relay constraint is formalized is Eq (3). We collect the notations used in Sec. II and in Sec. III in Tab. I.

$$\boldsymbol{B} \cdot \boldsymbol{x} \leq \boldsymbol{0} \tag{3}$$

A cyclic mesh structure is common in P2P applications, mostly in downloading applications where data relaying is coordinated in receiver-driven mode on a piece-by-piece basis, e.g., BitTorrent. Many P2P TV applications, albeit supporting video streaming, function in the same manner. Most importantly, these applications are not concerned with video adaptation issues because all peers receive exactly the same content. However, our solution is designed towards streaming scenarios in which adaptation can help such as the presence of network heterogeneity (streaming to handheld device) or strong real-time requirement overshadowing quality (multi-party conferencing). It is not uncommon to see a DAG distribution structure in these scenarios. For example, in a live broadcasting event, if a new peer always find its parent(s) among the current online peers, a DAG structure will form where the new peer can be sequenced according to their joining time.

III. Rate Adaptation

In this section, we present our rate allocation function that minimizes the aggregate rate distortion for all peers in the network. We present the rate distortion function as a performance evaluation metric for our solution. We then present a distributed rate allocation algorithm that minimizes rate distortion among all peers.

A. Performance Evaluation

The Mean-Squared-Error (MSE) is a commonly used metric for evaluation of video transmission. However, representing simulation results in PSNR is widely accepted in the multimedia coding and networking community. The PSNR is defined as

$$PSNR = 10\log_{10}(255^2/D) \tag{4}$$

where D represents the overall MSE of the entire encoded video sequence. For live video streaming, decoded video quality at the receiver is affected by distortions due to encoder compression and packet loss or late arrival of packets. Assuming that these distortion effects do not correlate [7], our experiment only focuses on the distortion due to the encoder compression. For a flow f with rate x_f, we adopt a parametric rate distortion function:

$$D_f(x_f) = \frac{\theta^s}{x_f - x_0^s} + D_0 \tag{5}$$

where the variables (θ, x_0 and D_0) depends on the encoded sequence and the percentage of INTRA coded macroblocks β. In our experiment, these values were fitted for each video with empirical data based on the trial encoding method described in [7].

B. Problem Formulation

We seek to minimize the aggregated distortion of the streaming video received by all peers. We adopt the rate distortion definition in Eq. (5) into a non-linear optimization problem that optimizes all the incoming flow rate of a peer. The objective function[2] is given as:

$$\begin{aligned} \textbf{min.} \quad & \sum_{h\in\mathcal{H}} \sum_{f\in\mathcal{R}(h)} D_f(x_f) \qquad (6) \\ \textbf{subject to} \quad & (1) \textbf{ and } (3) \\ \textbf{over} \quad & \boldsymbol{x} \in I_f \end{aligned}$$

where equation (6) is a convex function of the allocated rate. The rate of each flow x_f is adapted within the range $I_f = [m_f, M_f]$.

By non-linear optimization theory, there exists a minimizing value of rate vector $\boldsymbol{x}$ for the above optimization problem. We consider the Lagrangian form of this problem:

$$\begin{aligned} & L(\boldsymbol{x}, \boldsymbol{\mu}^\alpha, \boldsymbol{\mu}^\beta) \qquad (7) \\ = & \sum_{h\in\mathcal{H}} \sum_{f\in\mathcal{R}(h)} D_f(x_f) + \boldsymbol{\mu}^\alpha(\boldsymbol{A}\cdot\boldsymbol{x} - \boldsymbol{c}) + \boldsymbol{\mu}^\beta(\boldsymbol{B}\cdot\boldsymbol{x}) \end{aligned}$$

where $\boldsymbol{\mu}^\alpha = (\mu_l^\alpha,\ l \in \mathcal{L})$ and $\boldsymbol{\mu}^\beta = (\mu_k^\beta,\ k = 1,\ldots,H^2)$ are vectors of Lagrangian multipliers. Here μ_l^α is the *link price*. $\mu^\beta_{(h_i-1)H+h_k}$ is the *relay price* that peer h_k has to pay to its parent h_i for relaying the data.

[2]For simplicity purposes, we assume that the objective function excludes the rate distortion function for the server h_0. This can be easily achieved by assigning the value 0 to the rate distortion function of h_0.

Solving the problem (6) now involves two sets of prices, each corresponding to one of the two constraints defined in inequalities (1) and (3). Therefore, we give our solution to this problem the term double-pricing solution in contrast to the single-pricing solution that appeared in existing literatures [3], [4], [8]. Single-pricing solution only considers the capacity constraint (1) by treating all flows as independent from each other. Comparison of these two solutions will be the main theme of our simulation study.

C. Non-Linear Optimization

Problem (7) can be solved in a distributed way following the gradient projection method. The optimal rate x_f for a flow f can then be calculated as

$$x_f(\boldsymbol{\mu}^\alpha, \boldsymbol{\mu}^\beta) = \left[x_0^s + \sqrt{\frac{\theta^s}{\lambda_f^\alpha + \lambda_f^\beta}}\right]_{m_f}^{M_f} \tag{8}$$

Here $\boldsymbol{\lambda}^\alpha = (\lambda_f^\alpha, f \in \mathcal{F})$ and $\boldsymbol{\lambda}^\beta = (\lambda_f^\beta, f \in \mathcal{F})$ are two new vectors. λ_f^α is the sum of the prices of all links in the path of f, i.e., the *network price* mentioned in Eq. (1) that f has to pay. In the star topology, f only has to pay the price for the uplink of its sending peer. λ_f^β can be interpreted as the *relay price* for f which is the difference between the aggregated relay price of parent of $h(f)$ ($\sum_{h\to h(f)} \mu^\beta_{(h-1)H+h(f)}$) and the relay benefit h_f receives from all its children ($\sum_{h(f)\to h} \mu^\beta_{(h(f)-1)H+h}$).

D. Distributed Algorithm

We present a rate allocation algorithm based on our previous work [5] that is adopted for multi-path scenario [9]. Tab. II presents the distributed algorithm that each peer executes to update its rate.

The algorithm proceeds in rounds denoted as $t = 1, 2, \cdots$. During each round, a peer h_i calculates the rates of its incoming flows. For each flow $f \in \mathcal{R}(h_i)$, we calculate the link price μ_l^α for each link l associated with the flow and the relay price μ_f^β. After calculating the network price λ_f^α and the net relay price λ_f^β for each flow f, a peer then determines the minimum price among all the incoming flows and uses this minimum price to calculate the sum of its incoming flow rates by taking the inverse of the first derivate of D using Eq. (8). In the last two steps of the algorithm, the rate of each flow in $\mathcal{R}(h_i)$ is then adjusted based on this minimum price. While the flows with higher prices will have their rates reduced, the flows with the minimum price will have their rates increased. The goal is to have equal price among all incoming flows for each receiver. This ensures that at optimality, the prices of each incoming flow of the receiver is minimum and the flow rate is optimum [9].

E. Discussion

We now discuss the implementation of our algorithm in a distributed fashion. Our implementation is receiver based, i.e., receiver of a flow is the owner of that flow. First we start with the price update. To update the price of link l, one needs to know about its old price and rate of all flows going through it. Since we assume a star topology, l must be the uplink of a peer and all flows on this link must be generated from this peer as well. Therefore, peer owning l maintains the price μ_l^α. Since receiver is the owner of a flow, a peer needs to receive message about the incoming rate from all its parents to calculate the relay price μ_f^β.

To update the rate, we must calculate λ_f^α and λ_f^β. The receiver of a flow receives μ_l^α from all of its parents and children and calculate λ_f^α. It also receives μ_f^β from its children and calculate λ_f^β for all of its incoming flows. Determing the optimum rate does not require further message passing between a parent and children.

The price update of our algorithm requires no messaging overhead since all the computations are done locally. The flow rate update process needs rate information from its parent and its child peers. Since such a message can be blended into existing traffic between parent and children peers such as heartbeat message or acknowledgment message in transmission protocol (e.g., TCP or RTCP), the messaging overhead can be reduced significantly.

TABLE II
DISTRIBUTED RATE ALLOCATION ALGORITHM

Each end host h_i at times $t = 1, 2, \cdots$

Initialization

$\mathbb{F}_i = \{f \mid f \in \mathcal{R}(h_i)\}$

$\mathbb{R}_i = \emptyset$

for each $f \in \mathbb{F}_i$

  for each $l \in \mathcal{L}(f)$

    $\mu_l^\alpha(t+1) \leftarrow \left[\mu_l^\alpha(t) + \gamma\Big(\sum_{f\in\mathcal{F}(l)} x_f(t) - c_l\Big)\right]^+$

  end for

  $\lambda_f^\alpha(t) \leftarrow \sum_{l\in\mathcal{L}(f)} \mu_l^\alpha(t)$

  $\mathcal{R}(h_j) = \{f_{kj} \mid \forall k, h_k \rightarrow h_j, f = f_{ji}\}$

  $\mu_f^\beta(t+1) \leftarrow \left[\mu_f^\beta(t) + \gamma\Big(x_f(t) - \sum_{f^p\in\mathcal{R}(h_j)} x_{f^p}(t)\Big)\right]^+$

  for each $f_{ij} \in \mathcal{S}(h_i)$

    $\mu_{f_{ij}}^\beta(t+1) \leftarrow \left[\mu_{f_{ij}}^\beta(t) + \gamma\Big(x_{f_{ij}}(t) - \sum_{f\in\mathcal{R}(h_i)} x_f(t)\Big)\right]^+$

  end for

  $\lambda_f^\beta(t) \leftarrow \left[\mu_f^\beta(t) - \sum_{f_{ij}\in\mathcal{S}(h_i)} \mu_{f_{ij}}^\beta(t)\right]$

end for

$\lambda_{f'}^\alpha(t) = \min_{f\in\mathcal{R}(h_i)} \lambda_f^\alpha(t)$

$\lambda_{f'}^\beta(t) = \min_{f\in\mathcal{R}(h_i)} \lambda_f^\beta(t)$

$\lambda_{f'}(t) \leftarrow \lambda_{f'}^\alpha(t) + \lambda_{f'}^\beta(t)$

Source rate $x_{h_i}(t+1) \leftarrow \left[D'^{-1}(\lambda_{f'}(t))\right]_{m_f}^{M_f}$

for each $f \in \mathcal{R}(h_i)$ update flow rate

  $x_f(t+1) \leftarrow \left[x_f(t) - \gamma\Big(\lambda_f^\alpha(t) + \lambda_f^\beta(t) - \lambda_{f'}(t)\Big)\right]^+$

end for

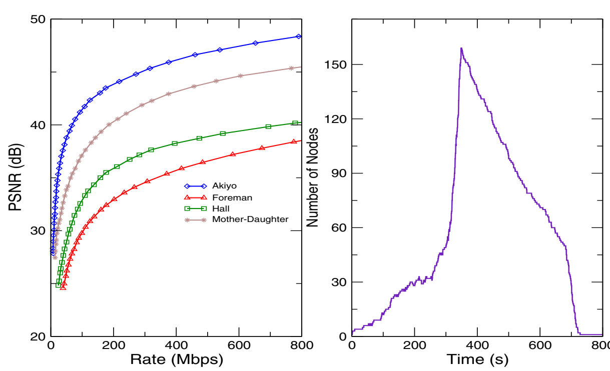
for each $f' \in \mathcal{R}(h_i)$ with minimum price $\lambda_{f'}$:

  $x_{f'}(t+1) \leftarrow \left[x_{h_i}(t+1) - \sum_{f\in\mathcal{R}\setminus f'} x_f(t+1)\right]^+$

end for

IV. SIMULATIONS

In this section, we present our simulation results. We start by introducing the setup of our simulation environment and elaborate on the mesh network construction. We then present our simulation results that compare the double-pricing and single-pricing solutions.



(a) (b)

Fig. 2. (a) Video transcoded using x264 [10] (b) Number of nodes joining and leaving the network

A. Video Input

We use transcoding to adapt our videos. In transcoding, we require each relaying peer to be able to adapt the quality of the video to fit into the receiving rates of its children peers. In our experiment, each peer performs transcoding by adjusting the quantization value of the video. In order to maximize the PSNR value for all the peers, our algorithm for rate assignment via transcoding chooses the highest quantized rate that is less than the receiving rate calculated by our algorithm. Formally, let x_{lf} be the optimal receiving rate for flow f calculated by our distributed algorithm, we denote $x_q = \{x_q \mid x_q < x_{q+1}, 1 \leq q \leq 51\}$ as the video encoding rate with quantization value q. The actual relay rate will then be x_f, where $x_f = \{x_q \mid x_q \leq x_{lf} < x_{q+1}\}$. We use the open-source software X264 [11] to encode videos with different quantization values. The test sequences used for our transcoding experiment are the ITU-T test sequences [10] *foreman*, *akiyo*, *hall* and *mother-daughter*, each having 300 frames with CIF resolution. Their PSNR vs. Rate are given in Fig. 2(a).

B. Mesh Construction

Here we present a simple mesh construction procedure in order to test our algorithm. However, our solution is topology independent and works with any mesh construction solution. We use real-time MSN video trace data [12] to construct the mesh. The traces provide the join/leave time stamp and the uplink bandwidth of each individual peers. We assign the server bandwidth to 2Mbps. We assume that a new update round begins every 0.1 second, i.e., members update thei flow rate at every 0.1 second. For the sake of simplicity, if there is more than 1 peer joining the network at any second, we queue them and add peers in the network at the beginning of each round. When a new peer wishes to join the network, we use the spare capacity information of the existing peers to determine a suitable parent. Formally, For each peer $h \in \mathcal{H}$ with link $l \in \mathcal{L}$, we define a *spare-capacity coefficient*[3] as:

$$s_h = [c_l - \sum_{f \in \mathcal{F}(l)} x_f]_0^{x_{\mathcal{R}(h)}} \quad (9)$$

where $x_{\mathcal{R}(h)}$ is the total incoming flow rate of peer h. The peers also calculate their *link-ratio*[4] to decide if they should seek multiple parents. Formally, this ratio is defined as:

$$r_h = \frac{x_{\mathcal{R}(h)}}{x_{\mathcal{S}(h)}} \quad (10)$$

Here $x_{\mathcal{S}(h)}$ defines the total outgoing rate. A peer having a $r_h < 1$ implies that it dedicates more bandwidth to its children than it receives. In order to address this imbalance, a peer seeks a parent with $r_h > 1$, a peer that receives more than it gives. During the simulation, we ensure that this process does not create a cycle. We also limit the maximum number of incoming/outgoing flows to 4. However, our solution is independent of the number of flows a peer can have.

At the end of each rate update cycle, child peers send the spare-capacity coefficient information and the link-ratio information to their parents. A parent then decides the best candidate with the highest coefficient value and sends this information along with the link-ratio to its own parent. The ID of the candidate with highest spare capacity eventually propagates to the server. The server also receives link-ratio information of all the peers. The server then provides a new peer with the parent information, so it can join the network as a child of that parent. It also provides existing peers with the ID of peers with $r_h > 1$ such that it does not create a cycle.

However, assigning a parent for a new peer based on the spare-capacity coefficient or for an existing peer based on the link-ratio may lead to different overlay mesh configuration for the single and double pricing solutions. In order to ensure fairness when comparing these two solutions, we use the same mesh. We apply the aforementioned multicast mesh construction criteria to construct the multicast mesh for the single pricing solution. The double pricing solution then replicates this configuration. The following procedures formally summarize the mesh construction process:

- **Rate Update**: At the end of each flow rate update, a peer updates its link-ratio and spare-capacity coefficient, i.e., the maximum rate it can provide to a new child. Peers then send this information to their parent.
- **ID Peer**: If child peers exist, a parent peer decides the best candidate based on its own spare coefficient and the spare coefficient of its children. It then sends this information to its own parent. This information eventually reaches the server.
- **Join**: The new peer sends a join request to the server. After receiving the parent information, the new peer joins the mesh as a child of the peer that maximizes its own receiving rate.
- **Req. Parent**: A peer wishing to have more than one parent sends a request through its parent to the server. The server then replies with an appropriate parent ID.

[3]The upper-bound in Eq.(9) does not apply to the server h_0 as its s_h is the difference between its capacity and the sum of its child flow rates

[4]The server has a link-ratio of 0. It can be thought of as the seed in a torrent network

C. Simulation Results

After trial and error, we adjust the initial value of μ_l^α and μ_f^β to 0.5 and the initial rate to 0.7Mbps. For our simulation, we use a step size (γ) of 0.3. Fig. 2(b) shows the number of nodes in the network at any point in time over an 800 second interval. As can be seen in Fig. 2(b), new peers stop joining the network at approximately 400 seconds and eventually all the peers leave the network. Fig. 3 shows the average PSNR value for the transcoded video sequences during that interval. Fig. 4 show the average rate over all the flows in the network. The rate and the PSNR value drops to 0 around 700 seconds, as the number of peers (excluding the server) in the network become 0. The results show that the double-pricing solution consistently performs better than the single-pricing solution.

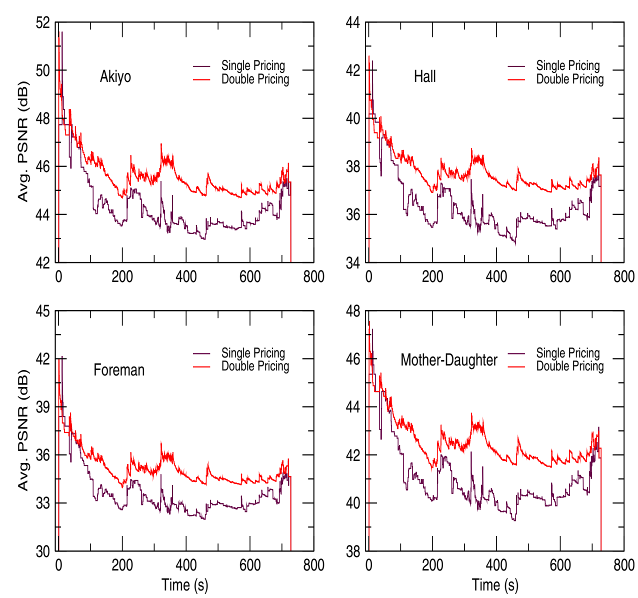


Fig. 3. PSNR value comparison of the double-pricing and single-pricing solutions for transcoded video

V. Related Work

Our optimization framework extends from the seminal works by Kelly [3], [4]. In this framework, the network resource allocation problem is formulated to maximize user utility under resource constraints. Price-based resource allocation strategies have been studied in the context of IP-unicast and multicast. Low et al. [8] extended this pricing strategy to a distributed algorithm. This price-based approach has also been applied to multirate multicast by Kar et al. [13] by using sub-gradient projections. Other works include multipath unicast [14] and streaming [15], multicast over wireless network [16], etc. A comprehensive review can be found in [17]. Our previous study applies the same framework to overlay multicast [5], which this paper is extended upon. However, based on [9], we extend the solution to a multi-path mesh network. In this paper, the target of our optimization framework is to minimize rate distortion combined with video adaptation techniques.

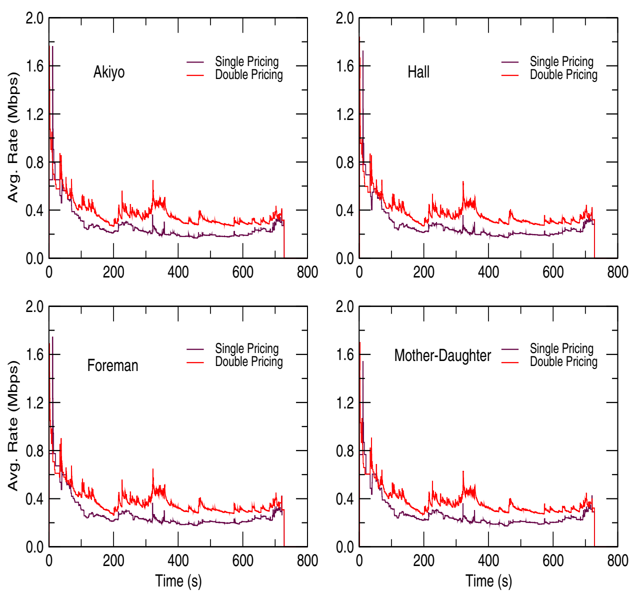


Fig. 4. Rate comparison of the double-pricing and single-pricing solution

Recently, this framework has been applied to multicast tree construction in P2P systems [6], another important area in P2P research. A long line of works focus on different routing structures including single tree [1], multiple trees [18], and mesh [2], etc. Our work deliberately avoids the routing functionality and instead focuses on the optimized rate allocation within any given routing structure.

The rate distortion function used in our optimization framework is proposed in [7] using a parametric model. There also exists many analytic models [19] to describe rate distortion. Finally, Hsu et al. [20] provides a comprehensive review on this subject.

VI. Conclusion

In this paper, we present an optimal rate allocation solution for multi-path P2P networks. We use non-linear optimization framework to minimize the aggregated distortion and thus maximize the overall PSNR among all peers in a P2P network. We consider the peer relaying price which is unique to a P2P distribution scenario along with the network price. Simulation shows that using the double pricing solution improves the aggregate rate distortion for all peers in the network and provides a better video experience compared to single pricing solution.

VII. Acknowledgement

The authors would like to thank Dr. Jin Li from Microsoft Research for providing MSN traces.

References

[1] Y. Chu, R. Rao, and H. Zhang, "A case for end system multicast," *In ACM SIGMETRICS*, 2000.
[2] N. Magharei, R. Rejaie, and Y. Guo, "Mesh or multiple-tree: A comparative study of live p2p streaming approaches," in *IEEE INFOCOM*, 2007.
[3] F. Kelly, "Charging and rate control for elastic traffic," *European Transactions on Telecommunications*, vol. 8, no. 1, 1997.
[4] F. Kelly, A. Maullo, and D. Tan, "Rate control for communication networks: Shadow prices, proportional fairness and stability," *Journal of Operations Research Society*, vol. 49, no. 3, 1998.
[5] Y. Cui, Y. Xue, and K. Nahrstedt, "Optimal resource allocation in overlay multicast," *IEEE International Conference on Network Protocols*, 2003.
[6] M. Chen, M. Ponec, S. Sengupta, J. Li, and P. A. Chou, "Utility maximization in peer-to-peer systems," in *SIGMETRICS '08: Proceedings of the 2008 ACM SIGMETRICS international conference on Measurement and modeling of computer systems*. ACM, 2008.
[7] K. Stuhlmuller, N. Farber, M. Link, and B. Girod, "Analysis of video transmission over lossy channels," *Selected Areas in Communications, IEEE Journal on*, vol. 18, no. 6, pp. 1012–1032, 2000.
[8] S. Low and D. Lapsley, "Optimization flow control, i: Basic algorithm and convergence," *IEEE/ACM Transactions on Networking*, vol. 7, no. 6, pp. 861–874, 1999.
[9] W. Wang, M. Palaniswami, and S. Low, "Optimal flow control and routing in multi-path networks," *ELSEVIER Performance Evaluation*, vol. 52, pp. 119–132, 2003.
[10] "Xiph.org: Test media," *http://media.xiph.org/video/derf/*.
[11] "Videolan - x264," *http://www.videolan.org/developers/x264.html*.
[12] C. Huang, J. Li, and K. Ross, "Can internet video-on-demand be profitable?" *Proc. SIGCOMM*, 2007.
[13] K. Kar, S. Sarkar, and L. Tassiulas, "Optimization based rate control for multirate multicast sessions," *In IEEE INFOCOM*, 2001.
[14] H. Han, S. Shakkottai, C. Hollot, R. Srikant, and D. Towsley, "Multipath tcp: A joint congestion control and routing scheme to exploit path diversity in the internet," *IEEE/ACM Trans. Networking*, 2006.
[15] D. Jurca and P. Frossard, "Media flow rate allocation in multipath networks," *IEEE Transactions on Multimedia*, vol. 9, no. 6, 2007.
[16] X. Zhu, T. Schierl, T. Wiegand, and B. Girod, "Video multicast over wireless mesh networks with scalable video coding(svc)," *Visual Communications and Image Processing*, January 2008.
[17] M. Chiang, S. H. Low, A. R. Calderbank, and J. C. Doyle, "Layering as optimization decomposition:a mathematical theory of network architectures," in *Proc. IEEE*, no. 1, 2007.
[18] M. Castro, P. Druschel, A.-M. Kermarrec, A. R. A. Nandi, and A. Singh, "Splitstream: High-bandwidth content distribution in a cooperative environment," in *ACM SOSP*, 2003.
[19] M. Dai and D. Loguinov, "Analysis of rate-distortion functions and congestion control in scalable internet video streaming," in *ACM NOSSDAV*, 2003.
[20] C. Hsu and M. Hefeeda, "On the accuracy and complexity of rate-distortion models for fgs-encoded video sequences," *ACM Transactions on Multimedia Computing, Communications, and Applications*, vol. 4, no. 2, 2008.