

San Jose State University

From the Selected Works of Xiao Su

August, 2005

Optimizing Transmission Time of Scalable Coded Images in Peer-to-Peer Networks

Xiao Su, *San Jose State University*

Rod Fatoohi, *San Jose State University*

Tao Wang, *Synopsys, Inc*



Available at: https://works.bepress.com/xiao_su/14/

Optimizing Transmission Time of Scalable Coded Images in Peer-to-Peer Networks

Xiao Su and Rod Fatoohi
Computer Engineering Department
San Jose State University
San Jose, CA 95192, USA
email: xsu@email.sjsu.edu

Tao Wang
Synopsys, Inc.
700 E. Middlefield Road
Mountain View, CA 94043, USA
email: tao.wang@synopsys.com

Abstract

In this paper, we study how to optimize image transmission time in peer-to-peer networks, by considering the property of bitstreams generated by image coding algorithms. Images can be either scalable coded or non-scalable coded. Since transmission of non-scalable coded images has been well understood, we focus on transmission of scalable coded images in this paper.

Because scalable coding embeds lower bit-rate bitstreams into higher bit-rate bitstreams, there exists a many-to-one relationship between supplying and requesting peers. When a requesting peer asks for an image coded in a particular bit rate, multiple peers with the same image but coded in different bit rates can supply to the requesting peer. This enables us to assign the task of image transmission to multiple supplying peers, split the traffic among these peers, and then reduce overall transmission time. Therefore, when we transmit scalable coded images over peer-to-peer networks, it is important to design optimal peer assignment algorithms to minimize the overall transmission time for the requesting peer. In the paper, we first formally define the peer assignment problem, and then establish a sufficient condition on the optimality of peer assignment. Based on this condition, we propose an optimal peer assignment algorithm in continuous space (OPA-CS) and subsequently derive a sub-optimal peer assignment algorithm in integer space (SOPA-IS). Finally, we carry out extensive experiments to verify the superior performance of the proposed peer assignment algorithms by comparing with two simple heuristic schemes.

1 Introduction

The delivery of multimedia content, such as audio, image and video, depends largely on two factors: (1) content delivery architecture; and (2) content coding algorithms that define the property of coded bit streams.

Compared with traditional client-server architecture, the peer-to-peer architecture is especially appealing to content delivery applications, as a requesting peer may obtain content from a set of less congested or geographically closer supplying peers. This makes these applications less susceptible to bandwidth shortage and network congestion [21].

Content coding algorithms can be coarsely classified into two categories: scalable coding [17, 18, 26] that embeds lower bit-rate bitstreams into higher bit-rate bitstreams, and non-scalable coding that does not have this embedding property. Traditional coding standards, such as JPEG [25] and MPEG-1 [7], typically generate non-scalable coded bitstreams, whereas newer standards, such as JPEG2000 [12] and MPEG-4 [19], typically generate scalable coded bitstreams.

As both peer-to-peer architecture and scalable coding standards are gaining popularity, it is essential to understand how to efficiently transmit scalable coded content over peer-to-peer networks. In this paper, we will focus on transmission of scalable coded images in such an architecture. To our best knowledge, this is the first effort along this line.

1.1 Background

The images are usually coded (*i.e.*, compressed) to reduce the storage space or network bandwidth, before they are stored in disk or transmitted over networks. The size of the bitstream, generated by coding an image, depends on coded bit rate r . For example, for an 512×512 image, if r is 0.5 bit per pixel (bpp), the size of the coded image is 128 kbit ($= 512 \times 512 \times 0.5/1024$). From now on, when we talk about transmission of images, we

mean transmission of coded images or coded bitstreams, instead of uncoded raw images.

The image coding standards include JPEG [25] that generates non-scalable coded images, and JPEG2000 [12] that generates scalable coded images. To understand how different coding algorithms affect image transmission on peer-to-peer networks, let us first elaborate the difference between scalable and non-scalable coding.

For this purpose, we represent the coded bitstream as a string of s bits $C = c_1c_2 \dots c_s$, where s is the size of the bitstream, increasing with coded bit rate r . Let C_1 and C_2 be the bitstreams generated by coding an image in bit rate r_1 and r_2 , respectively, and $r_1 < r_2$. Scalable coding generates C_1 as a prefix part of C_2 , denoted by $C_1 \prec C_2$, which is called embedding property, whereas non-scalable coding generates C_1 and C_2 as two entirely different strings.

Suppose peer 1 (p_1) and peer 2 (p_2) have C_1 and C_2 , respectively. A requesting peer asks for the image C_M of size M (coded in bit rate r_M) and $r_1 < r_M < r_2$. If the image is non-scalable coded, bitstreams C_1 , C_2 , and C_M are totally different, so both p_1 and p_2 are unable to supply their images to the requesting peer. However, if the image is scalable coded, bitstreams C_1 , C_2 , and C_M satisfy $C_1 \prec C_M \prec C_2$. Since $C_1 \prec C_M$, p_1 has the prefix subset of C_M and can supply a portion (or all) of C_1 to the requesting peer. Similarly, p_2 has the superset of C_M and can certainly supply a portion (or all) of C_2 to the requesting peer. In this case, both p_1 and p_2 can contribute to the requesting peer.

Therefore, the peer holding a non-scalable coded image can only supply to the requesting peer that asks for the same image of exactly the same bit rate. In contrast, the peer holding a scalable coded image can supply to the requesting peer that asks for the same image coded in a different bit rate.

To summarize, there are two important consequences of scalable coding on peer-to-peer (P2P) networks. First, more peers become eligible to serve as supplying peers, because those peers holding the requested image in different bit rates can contribute. This implies

a many-to-one relationship between the supplying and requesting peers, extending one-to-one correspondence found in early file sharing systems [3, 4, 6] that deal with non-scalable content. Second, the set of supplying peers is varying over time, because the supplying peers that have the images of different sizes and heterogeneous bandwidth may finish their own transmission at different times. Given this dynamic set of supplying peers, it is very important to investigate how to divide the requested scalable coded image (*e.g.* bitstream C_M) into image segments (in compressed domain) and how to assign these segments to the supplying peers (*e.g.* p_1 and p_2) in order to minimize image transmission time for the requesting peer, under the constraints that all the supplying peers have limited image sizes (*e.g.* C_1 and C_2) and their coded images have the embedding property (*e.g.* $C_1 \prec C_M \prec C_2$).

1.2 Peer-to-Peer Scalable Image Transmission System

In our peer-to-peer system, transmission of scalable coded images is done in three steps. First, a requesting peer employs a certain directory lookup algorithm to locate a potential set of supplying peers for a requested image. Second, the requesting peer applies a peer assignment algorithm to allocate image segments to the supplying peers with the objective to minimize the overall image transmission time. Third, the supplying peers are informed about their own allocations by the requesting peer and then start transmission.

We make the following assumptions on peer-to-peer systems with regard to scalable image transmission.

1. *Directory lookup.* There is a directory lookup scheme to return a complete list of potential supplying peers, which hold the requested scalable coded image in different sizes. To help this directory lookup service, each peer needs to register detailed information about its images, such as coding scheme, coded bit rate, image dimension, and so on. The directory lookup service can be centralized, fully distributed, or

employ efficient distributed hash table based techniques [15, 16, 20, 28].

2. *Supplying bandwidth.* Every peer in the system has a fair estimation of its outgoing bandwidth. As a peer normally does not open many connections simultaneously, it can easily estimate its bandwidth contributed to the image transmission request.
3. *Relationship in peers' bandwidth.* We consider a single requesting peer with incoming bandwidth B and multiple supplying peers with heterogeneous bandwidth b_1, b_2, \dots, b_n , where n is the number of the supplying peers. We assume that the sum of supplying bandwidth does not exceed the incoming bandwidth of the requesting peer, *i.e.*, $\sum_{i=1}^n b_i \leq B$. This can be easily achieved by dropping some supplying peers if the above condition is violated.
4. *Reliable delivery.* We assume images are transmitted using reliable transport protocols, such as TCP. In this case, the bandwidth refers to the effective bandwidth observed by the supplying peers using these protocols.

1.3 Related Work

Early commercial peer-to-peer file sharing systems, such as Napster [6] and Gnutella [4], normally identify a single supplying peer by its directory lookup algorithm and download files from this single peer. More recent systems, such as KaZaA [5], eDonkey [2] and BitTorrent [1], adopt a more general data sharing model, downloading media files from multiple sources. These systems generally treat the media files as regular data files, and they do not explore the properties and structures of coded bitstreams.

In research community, there have been a lot of efforts developing efficient algorithms for live media streaming on P2P networks. CoopNet [13, 14] proposed a framework to distribute media content to a potentially large population of hosts. It incorporates redundancy in network paths and media data to provide resilience to dynamic peer joins and depar-

tures. SplitStream [9] is a high-bandwidth content distribution system built on Pastry [16]. It studied how to evenly distribute traffic among all the participating nodes with different bandwidth capacities. In terms of multicast streaming, NICE [8] and ZIGZAG [23, 24] proposed algorithms and protocols to construct scalable application-level multicast for media streaming.

For on-demand streaming on P2P networks, Xu [27] proposed optimal media assignment algorithm (OTS_{p2p}) to minimize the initial buffering delay and also studied how to amplify the overall system capacity for media streaming. Cui [10, 11] exploited the buffer capacity at peer nodes to reduce the load on streaming servers when the user requests are asynchronous and the peers' bandwidths are heterogeneous.

In summary, previous research work has focused on various other aspects of peer-to-peer networks, and no work has been done to investigate how to optimize the delivery time for transmitting scalable coded images on P2P networks. Our work fills in this gap by exploiting the property of scalable image coding when designing transmission schemes.

1.4 Contributions of the Paper

The main contributions of this paper are summarized as follows. First, we exploit the property of scalable coded bitstreams in image transmission on peer-to-peer networks. This property makes more peers available as supplying peers, and thus enables us to assign the task of image transmission to multiple peers. By fully utilizing the bandwidth from the supplying peers, we can dramatically improve image transmission time. Second, we propose optimal peer assignment algorithms and theoretically establish the optimality of the algorithms. These algorithms are designed to address the challenge identified in P2P scalable image transmission. Third, we have conducted extensive simulations, and have verified excellent performance of the proposed algorithms.

1.5 Organization of the Paper

In Section 2, we formally define the peer assignment problem. In Section 3, we establish a sufficient condition on optimal peer assignment, based on which we propose an optimal peer assignment algorithm in continuous space (OPA-CS) and a sub-optimal peer assignment algorithm in integer space (SOPA-IS). To analytically assess the quality of this sub-optimal integer solution, we establish an upperbound on its distance from the optimal integer solution. We carry out extensive experiments to evaluate the performance of the proposed peer assignment algorithms in Section 4. Section 5 concludes the paper by identifying future research directions.

2 Problem Definition

Consider the following peer-to-peer image transmission system.

- For a given requesting peer, there are n supplying peers with image sizes s_i , coded in different bit rates r_i ($i = 1, 2, \dots, n$). Here s_i represents the size of scalable coded image held by peer i , whose bitstream is C_{s_i} (described in Section 1.1). Without loss of generality, we assume $s_1 \leq s_2 \leq \dots \leq s_n$, otherwise we can re-number the peers to follow this order. Hence the bitstreams satisfy the embedded property:

$$C_{s_1} \prec C_{s_2} \prec \dots \prec C_{s_n}.$$

- Supplying peer i has outgoing bandwidth b_i , $i = 1, 2, \dots, n$.
- The requesting peer asks for a coded image C_M (at bit rate r_M) of size M , which is less than or equal to the maximum image size, otherwise the request cannot be satisfied.

Given the above notations, let us define the following two concepts in the domain of coded images.

DEFINITION 1 Image allocation vector is defined as a partition of coded image C_M : $\{x_0, x_1, x_2, \dots, x_n\}$ with $x_0 = 0$ and $x_n = M$, so that the portion of the coded image between $(x_{i-1}, x_i]$ is assigned to peer i .

DEFINITION 2 Peer assignment vector $\{\Delta_i, i = 1, 2, \dots, n\}$ is the vector in which the i^{th} element, $\Delta_i = x_i - x_{i-1}$, defines the size of the coded image segment assigned to peer i .

Obviously, there is a one-to-one correspondence between an image allocation vector and a peer assignment vector: $x_i = \sum_{k=1}^i \Delta_k$ and $\Delta_i = x_i - x_{i-1}$ for $i = 1, 2, \dots, n$. In addition, the peer assignment vector satisfies the following condition:

$$\Delta_1 + \Delta_2 + \dots + \Delta_n = M \quad (1)$$

Based on the above definitions, image transmission time t (also called downloading time) is calculated as

$$t = \max\{\Delta_i/b_i, \quad i = 1, 2, \dots, n\} \quad (2)$$

The goal of a peer assignment algorithm is to find a peer assignment vector (or an image allocation vector) to minimize t .

Let us walk through a simple example to understand the above notations and how different peer assignment solutions affect image transmission time. In this example, a requesting peer asks for an image of size 200 kbit from three supplying peers, p_1 , p_2 , and p_3 . The three peers hold coded images of size 100 kbit, 175 kbit and 200 kbit, and supply them using bandwidth 20 kbit, 50 kbit, and 10 kbit per second, respectively. Next let us compare two peer assignment solutions.

In solution one, p_1 is assigned to transmit the image between $(0, 20]$ kbit, p_2 to transmit between $(20, 100]$ kbit, and p_3 to transmit between $(100, 200]$ kbit. The corresponding image allocation vector is $\{x_0 = 0, x_1 = 20, x_2 = 100, x_3 = 200\}$ and peer assignment vector is

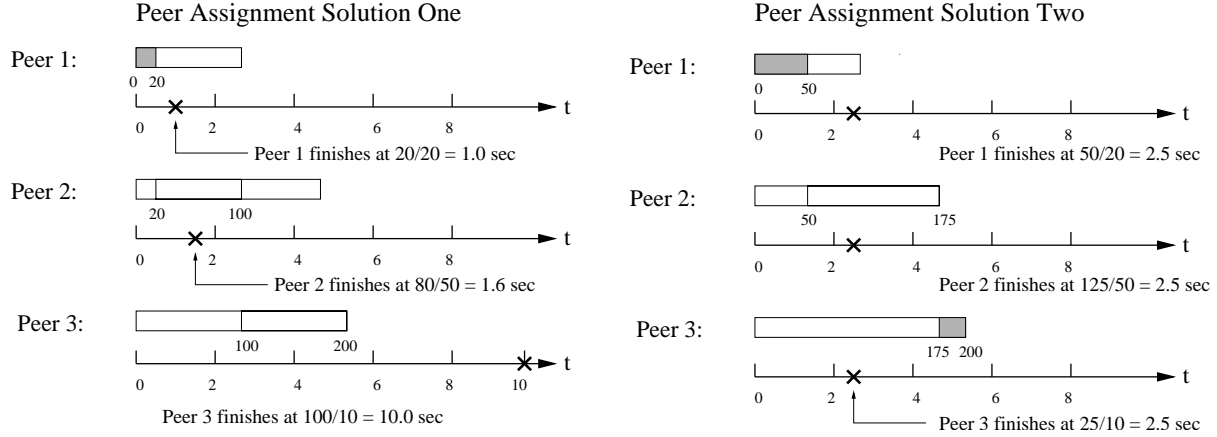


Figure 1: Comparisons of two peer assignment solutions.

$\{x_1 = x_1 - x_0 = 20, x_2 = x_2 - x_1 = 80, x_3 = x_3 - x_2 = 100\}$. As a result, the downloading time is equal to $\max\left\{\frac{20}{20}, \frac{80}{50}, \frac{100}{10}\right\} = 10.0$ seconds.

In solution two, p_1 is assigned to transmit between $(0,50]$ kbit, p_2 to transmit between $(50,175]$ kbit, and p_3 to transmit between $(175,200]$ kbit. This corresponds to an image allocation vector $\{x_0 = 0, x_1 = 50, x_2 = 175, x_3 = 200\}$ and a peer assignment vector $\{x_1 = x_1 - x_0 = 50, x_2 = x_2 - x_1 = 125, x_3 = x_3 - x_2 = 25\}$. In this case, the downloading time is equal to $\max\left\{\frac{50}{20}, \frac{125}{50}, \frac{25}{10}\right\} = 2.5$ seconds.

Clearly solution two has much shorter image transmission time than solution one. As a comparison, let us compute the image transmission time needed by existing P2P file downloading applications [4, 6], which do not explore the property of scalable coding. In this case, only p_3 can work as a supplying peer in the above example, resulting in image transmission time of 20 seconds. Therefore, considering the scalable coding property in image transmission can dramatically reduce downloading time from 20 seconds to 2.5 seconds.

Our objective in this paper is to design an optimal peer assignment algorithm to minimize image transmission time t .

3 Peer Assignment Algorithms

In this section, we first propose an optimal peer assignment solution when the image allocation vector takes continuous values, and then derive a sub-optimal peer assignment solution when the image allocation vector takes integer values. Last, we analytically assess the quality of this sub-optimal integer solution.

3.1 Optimal Peer Assignment in Continuous Space (OPA-CS)

To derive an optimal peer assignment vector, we observe that the transmission time is minimized if the aggregate bandwidth of all the supplying peers is maximally utilized, which means that all the supplying peers would start and finish transmission at the same time. If this is not true, and peer i takes longer than the rest of peers to finish, then during the time that only peer i is transmitting, we are only utilizing bandwidth b_i , instead of the aggregate bandwidth $b_1 + b_2 + \dots + b_n$. Based on the above analysis, Theorem 1 establishes a *sufficient condition* for the peer assignment vector.

THEOREM 1 *If the peer assignment vector $\{b_i, i = 1, 2, \dots, n\}$ satisfies the following condition:*

$$\frac{1}{b_1} = \frac{2}{b_2} = \dots = \frac{n}{b_n} \quad (3)$$

then image transmission time t , defined in Eq.(2), is minimized.

PROOF: If there exists an optimal peer assignment vector $\{b'_i, i = 1, 2, \dots, n\}$ that does not satisfy equality condition (3), then we sort the peers in the ascending order of their transmission time as follows:

$$\frac{1}{b'_1} \quad \frac{2}{b'_2} \quad \dots \quad \frac{n}{b'_n}$$

Define t and t as the image transmission time resulted from $\{b_i, i = 1, 2, \dots, n\}$ and $\{b_{p(i)}, i = 1, 2, \dots, n\}$, respectively, then we have

$$t = \frac{1}{b_1} = \frac{2}{b_2} = \dots = \frac{n}{b_n} \quad \text{and} \quad t = \frac{n}{b_n}$$

Let $p(\cdot)$ be the permutation function on the peer indices so that $b_{p(i)} = b_i, i = 1, 2, \dots, n$, then image transmission time t from $\{b_i, i = 1, 2, \dots, n\}$ can be rewritten as:

$$\begin{aligned} t &= \frac{p(1)}{b_{p(1)}} = \frac{p(2)}{b_{p(2)}} = \dots = \frac{p(n)}{b_{p(n)}} \\ &= \frac{p(1)}{b_1} = \frac{p(2)}{b_2} = \dots = \frac{p(n)}{b_n} \end{aligned}$$

By assumption, we have $t > t$, which implies

$$\frac{p(1)}{b_1} = \frac{p(2)}{b_2} = \dots = \frac{p(n)}{b_n} > \frac{n}{b_n} \geq \frac{n-1}{b_{n-1}} \geq \dots \geq \frac{1}{b_1}$$

The above equation leads to the following relationships:

$$t = \frac{p(i)}{b_i} > \frac{i}{b_i} \implies p(i) > i, \quad i = 1, 2, \dots, n \quad (4)$$

From Eq.(4), we can derive the following inequality between the two assignment vectors:

$$p(1) + p(2) + \dots + p(n) > 1 + 2 + \dots + n \quad (5)$$

However, Eq.(5) certainly contradicts with the fact that the total size of assigned image segments shall be equal to the requested file size M for both assignment solutions, as specified in Eq.(1). ■

Based on Theorem 1 and Definition 2, we have the following relationship on the image allocation vector:

$$\frac{x_1}{b_1} = \frac{x_2 - x_1}{b_2} = \frac{x_n - x_{n-1}}{b_n}$$

Given that $x_n = M$, we can easily derive:

$$x_i = \frac{\sum_{k=1}^i b_k}{\sum_{k=1}^n b_k} M, \quad i = 1, 2, \dots, n-1, \quad \text{and} \quad x_n = M \quad (6)$$

If Eq.(6) satisfies the following boundary constraint that ensures the assigned portion for a given peer is a sub-part of the coded image for that peer,

$$x_i \leq s_i, \quad i = 1, 2, \dots, n-1, \quad (7)$$

then we have found an optimal peer assignment solution. However, in most cases, x_i 's derived from Eq.(6) do not satisfy boundary constraint in Eq.(7). Therefore, we need to develop an algorithm to consider this constraint.

Figure 2 outlines the optimal peer assignment algorithm in continuous space (OPA-CS), taking into account boundary constraint Eq.(7). The basic principle underlying the algorithm is that if peer i cannot fulfill its assigned portion, we need to re-allocate the missing part δ_i to as many remaining peers as possible. The algorithm works as follows. Start with peer 1 and calculate its assigned portion according to Eq.(6). If this assignment is part of the image that peer 1 has, then proceed to peer 2, otherwise, re-allocate δ_1 , the out-of-bound part of peer 1, among peers 2, 3, \dots , n according to Eq.(6), treating δ_1 as a new image transmission request. This process continues until the last peer.

Next, we prove that OPA-CS achieves the minimum transmission time when x_i 's take continuous values, regardless of whether peers satisfy the boundary constraint in Eq.(7).

THEOREM 2 *OPA-CS minimizes transmission time t if $x_i \in R^+ \cup \{0\}$.*

$$\begin{aligned}
(1) \quad x_1^0 &= \frac{b_1}{\sum_{k=1}^n b_k} M \\
x_1 &= \min(x_1^0, s_1) \\
\delta_1 &= \begin{cases} x_1^0 - s_1 & \text{if } x_1^0 > s_1, \\ 0 & \text{otherwise.} \end{cases} \\
(2) \quad x_2^0 &= \frac{b_1 + b_2}{\sum_{k=1}^n b_k} M + \frac{b_2}{\sum_{k=2}^n b_k} \delta_1 - \delta_1 \\
x_2 &= \min(x_2^0, s_2) \\
\delta_2 &= \begin{cases} x_2^0 - s_2 & \text{if } x_2^0 > s_2, \\ 0 & \text{otherwise.} \end{cases} \\
&\vdots \\
(i) \quad x_i^0 &= \frac{\sum_{k=1}^i b_k}{\sum_{k=1}^n b_k} M + \sum_{j=1}^{i-1} \left(\frac{\sum_{k=j+1}^i b_k}{\sum_{k=j+1}^n b_k} \delta_j - \delta_j \right) \\
x_i &= \min(x_i^0, s_i) \\
\delta_i &= \begin{cases} x_i^0 - s_i & \text{if } x_i^0 > s_i, \\ 0 & \text{otherwise.} \end{cases} \\
&\vdots
\end{aligned}$$

Figure 2: Optimal peer assignment in continuous space (OPA-CS).

PROOF: Let us represent the peer assignment vector resulted from OPA-CS as $\{x_1, x_2, \dots, x_n\}$.

Then our objective is to show that image transmission time, $t = \max\{\frac{\Delta_1}{b_1}, \frac{\Delta_2}{b_2}, \dots, \frac{\Delta_n}{b_n}\}$, is minimized.

Let us denote the peer assignment vector calculated from Eq.(6) (*i.e.*, the ideal allocation without the boundary constraint) as $\{i_1, i_2, \dots, i_n\}$ and its transmission time as t_{eq} .

If $i_j = i_j$ for $j = 1, 2, \dots, n$, then $t = t_{eq}$. Based on the sufficient condition in Theorem 1, t is minimized, and we are done.

If $\frac{i_j}{b_j}$'s are not equal, then we have $t > t_{eq}$. Let $t = t_{eq} + \delta t$. For our purpose, it suffices to show that δt is minimized. In the following, we prove this by mathematical induction. Let m be the total number of peers with their image allocations exceeding their boundaries, and i_1, i_2, \dots, i_m be the indices of these peers, and $t_j, j = 1, 2, \dots, m$, denote the increase in transmission time due to the violation of the i_j^{th} peer's boundary constraint.

If $m = 1$, then there is only one peer that finishes before transmitting its assigned image portion. Let i_1 be the index of this peer, then $\delta_{i_1} > 0$. Using the initial image allocation based on Eqn.(6), these n supplying peers will only be able to transmit a sub-segment of the requested image, and the size of this sub-segment is $M - \delta_{i_1}$. The supplying peers will finish transmitting this sub-segment at time t_{eq} . However, the overall transmission will be longer than t_{eq} , because the missing segment δ_{i_1} also needs to be transmitted. Observe that the peers are ordered in increasing image size, so only those peers that have longer image size than peer i_1 (*i.e.*, $j > i_1$) are eligible to transmit δ_{i_1} . OPA-CS re-allocates δ_{i_1} to these peers based on Eqn.(6), where the image size is equal to δ_{i_1} and the number of peers is equal to $n - i_1$. Therefore, the increase in overall image transmission time t_1 is minimized. As a result, the claim holds true when $m = 1$.

Suppose $m = k$, the claim also holds true. In other words, the increase in overall image transmission time, $\delta t = t_1 + t_2 + \dots + t_k$, is minimized when there are k peers with image allocations exceeding their boundaries.

It is easy to see that when there are $k + 1$ peers with image allocations exceeding their image boundaries ($m = k + 1$), the increase in overall image transmission time $\delta t = t_1 + t_2 + \dots + t_k + t_{k+1}$ is also minimized. To establish this, we can similarly partition the requested image into two segments: one of size $M - \delta_{i_{k+1}}$ and the other of size $\delta_{i_{k+1}}$. Based on the assumption of $m = k$, the time to transmit the first segment of size $M - \delta_{i_{k+1}}$ is minimized. The second segment is re-allocated to peers with indices $j > i_{k+1}$ based on Eqn.(6), therefore, t_{k+1} is also minimized. As a result, the claim holds true when $m = k + 1$.

In summary, we have proven that OPA-CS minimizes transmission time t if x_i 's take continuous values. ■

3.2 Sub-Optimal Peer Assignment in Integer Space (SOPA-IS)

In image transmission, x_i 's take non-negative integers in terms of bits or bytes. In this subsection, we propose an algorithm to round x_i (obtained from OPA-CS) to non-negative integers. The basic idea of this rounding algorithm is to minimize the increase of transmission time for each peer, compared with the optimal solution in continuous space.

let us denote $\{\hat{x}_1, \hat{x}_2, \dots, \hat{x}_n\}$ as an optimal continuous peer assignment vector, and $\{\hat{x}_1, \hat{x}_2, \dots, \hat{x}_n\}$ as its corresponding image allocation vector, and u as a basic rounding unit in terms of bits. Here, u is an integer whose value can be larger than or equal to one, depending on how images are scalable coded and transmitted using packets. For every \hat{x}_i , we can either round it to $\lfloor \hat{x}_i/u \rfloor u$ or $\lceil \hat{x}_i/u \rceil u$. The rounding criterion is to minimize the increase in the transmission time compared with the continuous solution. If we round \hat{x}_i to $\lfloor \hat{x}_i/u \rfloor u$, then the increase in transmission time is the time for peer $i + 1$ to transmit the portion between $\lfloor \hat{x}_i/u \rfloor u$ and \hat{x}_i . Similarly if we round \hat{x}_i to $\lceil \hat{x}_i/u \rceil u$, then the increase in transmission time is the time for peer i to transmit the portion between \hat{x}_i to $\lceil \hat{x}_i/u \rceil u$. Based on this analysis, we present our rounding algorithm in Figure 3.

1. Calculate $f_i = \hat{x}_i - \lfloor \hat{x}_i/u \rfloor u$, for $i = 1, 2, \dots, n$.
2. **foreach** i in $\{1, 2, \dots, n\}$ **do**
3. **if** f_i is equal to zero **then** do nothing and skip
4. **if** $\frac{f_i}{b_{i+1}} < \frac{u-f_i}{b_i}$
5. **then** $x_i = \lfloor \hat{x}_i/u \rfloor u$
6. **else** $x_i = \lceil \hat{x}_i/u \rceil u$
7. **end-for**

Figure 3: Rounding to integer solutions.

Note that this method makes the rounding decision based only on the transmission time of a single peer, regardless of other peers. Therefore, it is only a sub-optimal solution. To assess the quality of this rounded solution, we derive an upper bound on the distance between this rounded solution and the optimal integer solution.

THEOREM 3 *The difference between the transmission time of the rounded solution and that of the optimal integer solution is upper bounded by $\frac{u}{\min\{b_1, b_2, \dots, b_n\}}$, where u is the basic unit of rounding in terms of bits.*

PROOF: Let t_{cont} and t_{int} be the minimum transmission time for continuous and integer versions of peer assignment, respectively. Because the search space for integer optimal peer assignment is a subspace of that of the continuous peer assignment, the optimal continuous solution will be at least as good as the optimal integer solution. Therefore, we have the following relationship: $t_{cont} \leq t_{int}$.

Denote t_{round} as the transmission time obtained by our rounding algorithm, then we have $t_{round} - t_{int} \leq t_{round} - t_{cont}$.

Let $\{\hat{x}_1, \hat{x}_2, \dots, \hat{x}_n\}$ be an optimal continuous solution and $\{x_1 = \hat{x}_1 + \beta_1, x_2 = \hat{x}_2 + \beta_2, \dots, x_n = \hat{x}_n + \beta_n\}$ be its rounded integer solution obtained by the algorithm in Figure 3.

We first derive the bounds for β_i ($i = 1, 2, \dots, n$). From Line 4 in Figure 3, we know that

if $f_i < \frac{b_{i+1}}{b_i + b_{i+1}}u$, x_i is equal to $\lfloor \hat{x}_i/u \rfloor u$, resulting in $\beta_i = -f_i$, therefore, $\beta_i > -\frac{b_{i+1}}{b_i + b_{i+1}}u$.

Similarly, if $f_i \geq \frac{b_{i+1}}{b_i + b_{i+1}}u$, x_i is equal to $\lceil \hat{x}_i/u \rceil u$, resulting in $\beta_i = u - f_i$, therefore,

$\beta_i = \frac{b_i}{b_i + b_{i+1}}u$. So we have $\beta_i \in (-\frac{b_{i+1}}{b_i + b_{i+1}}, \frac{b_i}{b_i + b_{i+1}}]u$ for $i = 1, 2, \dots, n$. On the boundary, we set $\beta_0 = 0$.

Having obtained the range of β_i , we can derive the distance between the rounded solution and the optimal solution as follows.

$$\begin{aligned}
& t_{round} - t_{int} \quad t_{round} - t_{cont} \\
& \max \left\{ \frac{(\hat{x}_i + \beta_i) - (\hat{x}_{i-1} + \beta_{i-1})}{b_i} - \frac{\hat{x}_i - \hat{x}_{i-1}}{b_i}, i = 1, 2, \dots, n \right\} \\
& = \max \left\{ \frac{\beta_i - \beta_{i-1}}{b_i}, i = 1, 2, \dots, n \right\} \\
& \max \left\{ \left(\frac{b_i}{b_i + b_{i+1}} + \frac{b_i}{b_{i-1} + b_i} \right) u / b_i, i = 1, 2, \dots, n \right\} \\
& \frac{u}{\min\{b_1, b_2, \dots, b_n\}} \quad \blacksquare
\end{aligned}$$

The theorem implies that the higher the supplying bandwidth is, the smaller the bound is. As the bandwidth values, b_i 's ($i = 1, 2, \dots, n$), are in the range of kbit per second, this upper bound will be a very small fractional value. For example, if the rounding unit is one byte, and the minimum supplying bandwidth is 56 kbps, of a dialup modem, then the rounded solution is within 0.00014 ($\approx 8/56000$) second from the optimal solution.

4 Experimental Results

In this section, we evaluate the performance of our proposed peer assignment algorithms. First, we compare the quality of the optimal peer assignment algorithm in continuous space (OPA-CS) with the sub-optimal peer assignment algorithm in integer space (SOPA-IS) described in Section 3. Then we compare SOPA-IS with two simple heuristic peer assignment schemes.

In our experiments, we set the range of bandwidth to be between 64 byte/sec and 4 kbyte/sec, and consider the images of sizes 512×512 and 1024×1024 coded in 0.5 bpp (bit per pixel) and 1 bpp, respectively. Therefore, for an image of 512×512 , the size of a requested coded image is either 16 kbyte or 32 kbyte, and for an image of 1024×1024 , the requested image is either 64 kbyte or 128 kbyte. Since the supplying peers can have the image with size either less or greater than the requested image, we set the image size to be between $[4, 32]$ kbytes for 512×512 images and between $[16, 128]$ kbyte for 1024×1024 images.

For each of the requested image size, we perform the experiments for the peer-to-peer systems consisting of 2, 4, 8, 12, 16, 20 and 24 peers. The experiments are done on a Dell workstation with Pentium-III 1.8GHz CPU and 512M memory. All the reported results are calculated as the average of 100 runs.

4.1 Quality Comparison of OPA-CS and SOPA-IS

In this subsection, we compare the quality of OPA-CS that finds optimal continuous solutions with SOPA-IS that finds sub-optimal integer solutions. Tables 1 and 2 show the comparison results when the requested images are of size 512×512 and are coded in 0.5 bpp (*i.e.*, coded image size = 16 kbyte) and 1 bpp (*i.e.*, coded image size = 32 kbyte), respectively. Similarly Tables 3 and 4 show the results for the images of size 1024×1024 , which are coded in 0.5 bpp (*i.e.*, coded image size = 64 kbyte) and 1 bpp (*i.e.*, coded image size = 128 kbyte), respectively. From the comparison results, we see that the sub-optimal solutions found by SOPA-IS are very close to the optimal solutions by OPA-CS. In Section 3, we have derived an upper bound on the distance between these two solutions, which is equal to $\frac{u}{\min\{b_1, b_2, \dots, b_n\}}$, where u is set to 8 bits (1 byte) here. Therefore, the difference should be smaller than $1/64 = 0.015625$ second, and this is verified by these experiments.

Table 1: Comparison of two algorithms, OPA-CS and SOPA-IS, when the requested coded image size is equal to 16 kbyte. Time is measured in seconds.

#Peers	OPA-CS transmission time (t_c)	SOPA-IS transmission time (t_i)	comparison $t_i - t_c$
2	4.7381	4.7386	0.0005
4	2.2068	2.2076	0.0008
8	1.0372	1.0382	0.0010
12	0.6787	0.6800	0.0013
16	0.5055	0.5073	0.0018
20	0.4044	0.4059	0.0015
24	0.3329	0.3351	0.0022

Table 2: Comparison of two algorithms, OPA-CS and SOPA-IS, when the requested coded image size is equal to 32 kbyte. Time is measured in seconds.

#Peers	OPA-CS transmission time (t_c)	SOPA-IS transmission time (t_i)	comparison $t_i - t_c$
2	13.0606	13.0609	0.0003
4	6.9337	6.9344	0.0007
8	6.3474	6.3485	0.0009
12	5.8206	5.8216	0.0010
16	3.1332	3.1347	0.0015
20	2.8010	2.8020	0.0010
24	2.6412	2.6423	0.0011

Comparing the actual transmission time between 1 second and over 100 seconds, the difference is very small. In summary, we can conclude that SOPA-IS can find high-quality near-optimal solutions.

4.2 Comparison of SOPA-IS with Simple Heuristic Schemes

In this subsection, we study how SOPA-IS compares with other heuristic peer assignment schemes. Since no previous work was reported in the literature to do peer assignment for scalable coded images, we consider the following two simple heuristics for the purpose of comparisons.

Table 3: Comparison of two algorithms, OPA-CS and SOPA-IS, when the requested coded image size is equal to 64 kbyte. Time is measured in seconds.

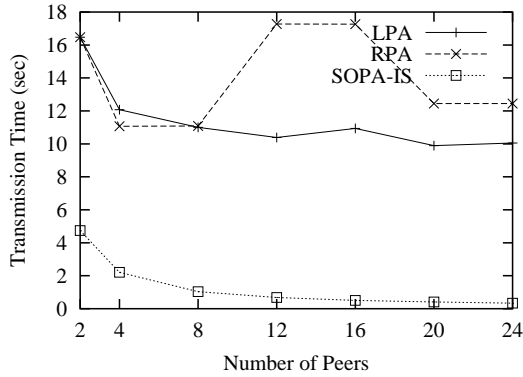
#Peers	OPA-CS transmission time (t_c)	SOPA-IS transmission time (t_i)	comparison $t_i - t_c$
2	19.7156	19.7160	0.0004
4	9.2848	9.2860	0.0012
8	4.1809	4.1820	0.0011
12	2.7558	2.7573	0.0015
16	2.0500	2.0517	0.0017
20	1.6249	1.6266	0.0017
24	1.3467	1.3487	0.0020

Table 4: Comparison of two algorithms, OPA-CS and SOPA-IS, when the requested coded image size is equal to 128 kbyte. Time is measured in seconds.

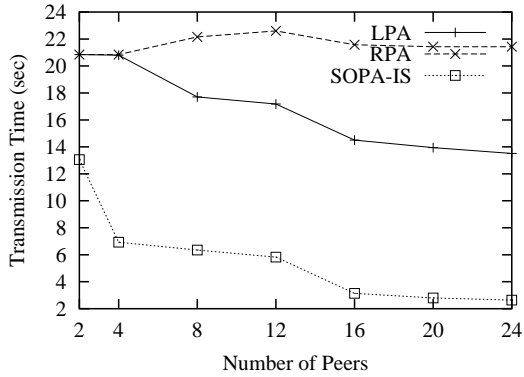
#Peers	OPA-CS transmission time (t_c)	SOPA-IS transmission time (t_i)	comparison $t_i - t_c$
2	113.5220	113.5220	0.0000
4	78.1452	78.1454	0.0002
8	33.3639	33.3644	0.0005
12	22.1170	22.1176	0.0006
16	6.1093	6.1107	0.0014
20	4.8479	4.8495	0.0016
24	4.1773	4.1789	0.0016

- Length-based peer assignment (LPA) in which peer i is assigned to transmit $\min\{s_i, M\} - s_{i-1}$ ($s_0 = 0$). In this scheme, the requesting peer only needs to do simple subtractions to come up with peer assignment.
- Random peer assignment (RPA) in which the requesting peer randomly selects a peer to transmit the portion from the current *lowerbound* to the size of this peer’s coded image. The variable *lowerbound* is initially set to zero and is updated to the image size of the selected peer after each iteration.

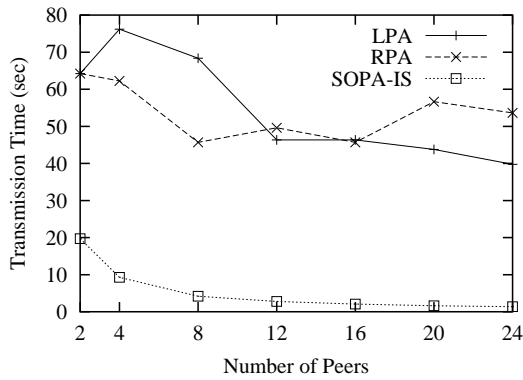
Figure 4 shows the comparison results when the size of a requested image is equal to 16 kbyte, 32 kbyte, 64 kbyte and 128 kbyte, respectively. It is not surprising that SOPA-IS



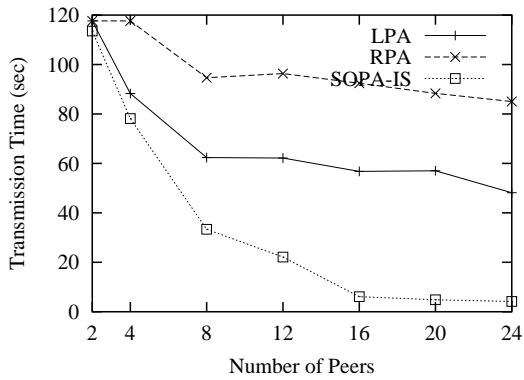
a) requested image size = 16 kbyte



b) requested image size = 32 kbyte



c) requested image size = 64 kbyte



d) requested image size = 128 kbyte

Figure 4: Performance comparisons of three peer assignment algorithms when image size is equal to a) 16 kbyte, b) 32 kbyte, c) 64 kbyte, and d) 128 kbyte, respectively.

outperforms the above two heuristics as SOPA-IS has been shown to produce near-optimal solutions. However, Figure 4 shows that SOPA-IS finds significantly better solutions than these two heuristics. As an example, in Figure 4a), SOPA-IS only takes from 3.3% to 28.8% of the transmission time needed by LPA, and from 2.7% to 28.8% of the transmission time needed by RPA.

In terms of computational time, LPA and RPA take about 0.1 milliseconds and SOPA-IS takes 1 to 2 milliseconds to complete. Since the image transmission time is in the order of seconds, the computational overhead of these three methods is negligible.

We can also observe that the transmission time of SOPA-IS decreases with increasing number of supplying peers, which is a desirable property of a good peer assignment algorithm. In contrast, both LPA and RPA do not demonstrate this property.

5 Conclusions and Future Work

As peer-to-peer architecture and scalable image coding are becoming popular, it is important to understand how to efficiently transmit scalable coded images over peer-to-peer networks. In this paper, we described our efforts to address this problem. We first defined the peer assignment problem, and then proposed OPA-CS and SOPA-IS. The quality of SOPA-IS has been shown to be very close to the optimal solution through both theoretical analysis and simulation results. In addition, we have also verified the superior performance of the proposed peer assignment schemes by comparing with two simple heuristics. Our proposed algorithms can reduce image transmission time up to two orders of magnitude compared to heuristic algorithms. The improvement is due to two key factors. First, we harvested more supplying peers by exploiting the property of scalable coding. Second, we designed an optimal assignment algorithm for image transmission.

Future work can be done in several directions. First, in this paper, we assumed that each supplying peer will complete its transmission assignment before leaving the system. In practice, this may not always be true, especially for mobile P2P systems. Peers may leave P2P network, crash, or move out of transmission range without notice. To design robust peer assignment schemes in such scenarios, we plan to incorporate error control techniques and statistical modeling of peers' life time into our algorithm. Second, scalable coded images can be displayed before we receive the entire coded image, so how to optimize display quality given a user-specified delay bound on transmission time is also an interesting problem. We have done some preliminary work [22] along this direction and are currently improving our algorithms. Third, we plan to investigate how to extend this work to scalable

video streaming, *e.g.*, videos coded by H.264 or MPEG-4. This would be of more practical significance, since video files are normally very large, and it takes much longer time to transmit videos than images. However, this is also a very challenging problem since video streaming has more stringent requirements on QoS, for example, small startup latency, continuous playback, and good visual quality.

References

- [1] BitTorrent. <http://bitconjurer.org/BitTorrent>.
- [2] eDonkey. <http://www.edonkey2000.com>.
- [3] FastTrack. <http://www.fasttrack.nu>.
- [4] Gnutella. <http://gnutella.wego.com>.
- [5] KaZaA. <http://www.kazaa.com/us/index.htm>.
- [6] Napster. <http://www.napster.com>.
- [7] ISO/IEC IS 11172-2 (MPEG-1 Video): Coding of moving pictures and associated audio for digital storage media at up to about 1.5 Mbit/s, 1993.
- [8] S. Banerjee, B. Bhattacharjee, and C. Kommareddy. Scalable application layer multicast. In *Proc. ACM SIGCOMM*, August 2002.
- [9] M. Castro, P. Druschel, A.-M. Kermarrec, A. Nandi, A. Rowstron, and A. Singh. SplitStream: high-bandwidth content distribution in a cooperative environment. In *Proc. IPTPS'03*, February 2003.
- [10] Y. Cui, B. Li, and K. Nahrstedt. oStream: asynchronous streaming multicast in application-layer overlay networks. *IEEE journal on Selected Areas in Communications*, 22:91–106, January 2004.
- [11] Y. Cui and K. Nahrstedt. layered peer-to-peer streaming. In *Proc. ACM/IEEE NOSS-DAV*, 2003.
- [12] International Standards Organization. JPEG 2000 image coding system. Final Committee Draft of ISO International Standard 15444 Part 1.
- [13] V. N. Padmanabhan, H. J. Wang, and P. A. Chou. Resilient peer-to-peer streaming. Technical Report MSR-TR-2003-11, Microsoft Research, March 2003.
- [14] V. N. Padmanabhan, H. J. Wang, P. A. Chou, and K. Sripanidkulchai. Distributing streaming media content using cooperative networking. In *Proc. ACM/IEEE NOSS-DAV*, Miami, FL, USA, May 2002.

- [15] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker. A scalable content-addressable network. In *Proc. ACM SIGCOMM*, pages 161–172, August 2001.
- [16] A. Rowstron and P. Druschel. Pastry: scalable, distributed object location and routing for large-scale peer-to-peer systems. In *Proc. IFIP/ACM Int'l Conf. on distributed systems platforms (Middleware)*, pages 329–350, Heidelberg, Germany, November 2001.
- [17] A. Said and W. A. Pearlman. A new fast and efficient image codec based on set partitioning in hierarchical trees. *IEEE Trans. on Circuits and Systems for Video Technology*, 6:243-250, June 1996.
- [18] J. M. Shapiro. Embedded image coding using zerotrees of wavelet coefficients. *IEEE Trans. on Signal Processing*, 41(12):3445-3462, December 1993.
- [19] T. Sikora. The MPEG-4 video standard verification model. *IEEE Trans. on Circuits and Systems for Video Technology*, 7(1):19-31, February 1997.
- [20] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan. chord: a scalable peer-to-peer lookup service for Internet applications. In *Proc. ACM SIGCOMM*, pages 149–160, August 2001.
- [21] D. Stolarz. Peer-to-peer streaming media delivery. In *Proc. of First Int'l Conf. on Peer-to-Peer Computing*, August 2001.
- [22] X. Su, Y. Shang, and Y. Mai. Delay-sensitive delivery of scalable coded images over peer-to-peer networks. In *Proc. IEEE Int'l Conf. on Multimedia and Expo*, pages 1627–1630, June 2004.
- [23] D. A. Tran, K. A. Hua, and T. T. Do. ZIGZAG: an efficient peer-to-peer scheme for media streaming. In *Proc. IEEE INFOCOM*, April 2003.
- [24] D. A. Tran, K. A. Hua, and T. T. Do. A peer-to-peer architecture for media streaming. *IEEE journal on Selected Areas in Communications*, 22:121–133, January 2004.
- [25] G. K. Wallace. The JPEG still picture compression standard. *IEEE Trans. on Consumer Electronics*, 38(1), February 1992.
- [26] Y. Wang, J. Ostermann, and Y.-Q. Zhang. *Video Processing and Communications*. Prentice Hall, NJ, 1st edition, 2001.
- [27] D. Xu, M. Hefeeda, S. Hambrusch, and B. Bhargava. On peer-to-peer media streaming. In *Proc. of SPIE/ACM Multimedia Computing and Networking*, San Jose, CA, January 2002.
- [28] B. Y. Zhao, L. Huang, J. Stribling, S. C. Rhea, A. D. Joseph, and J. D. Kubiatowicz. Tapestry: a resilient global-scale overlay for service deployment. *IEEE journal on Selected Areas in Communications*, 22:41–53, January 2004.