Optimal Bandwidth Assignment for Multiple-Description-Coded Video

Pengye Xia, S.-H. Gary Chan, *Senior Member, IEEE*, and Xing Jin

***Abstract*—In video streaming over multicast network, user bandwidth requirement is often heterogeneous possibly with orders of magnitude difference (say, from hundreds of kb/s for mobile devices to tens of Mb/s for high-definition TV). Multiple description coding (MDC) can be used to address this bandwidth heterogeneity issue. In MDC, the video source is encoded into multiple independent descriptions. A receiver, depending on its available bandwidth, joins different descriptions to meet their bandwidth requirements.**

An important but challenging problem for MDC video multicast is how to assign bandwidth to each description in order to maximize overall user satisfaction. In this paper, we investigate this issue by formulating it as an optimization problem, with the objective to maximize user bandwidth experience by taking into account the encoding inefficiency due to MDC.

We prove that the optimization problem is NP-hard. However, if the description number is larger than or equal to a certain threshold (e.g., if the minimum and maximum bandwidth requirements are 100 kb/s and 10 Mb/s, respectively, such threshold is seven descriptions), there is an exact and simple solution to achieve maximum user satisfaction, i.e., meeting all the bandwidth requirements. For the case when the description number is smaller, we present an efficient heuristic called simulated annealing for MDC bandwidth assignment (SAMBA) to assign bandwidth to each description given the distribution of user bandwidth requirement.

We evaluate our algorithm using simulations. SAMBA achieves virtually the same optimal performance based on exhaustive search. By comparing with other assignment algorithms, SAMBA significantly improves user satisfaction. We also show that, if the coding efficiency decreases with the number of descriptions, there is an optimal description number to achieve maximal user satisfaction.

***Index Terms*—Multiple-description-coded video, optimal description bandwidth assignment, simulated annealing, streaming.**

I. Introduction

With the penetration of broadband Internet access and advances in video compression techniques, there has been increasing interest in both stored and live video services. Websites like YouTube and MSN Video have offered numerous on-demand video clips. Online live TV streaming with the use of IP multicast or peer-to-peer (P2P) technology have also been widely deployed (e.g., AT&T IPTV, PPLive, CoopNet, and SplitStream) [1]–[4].

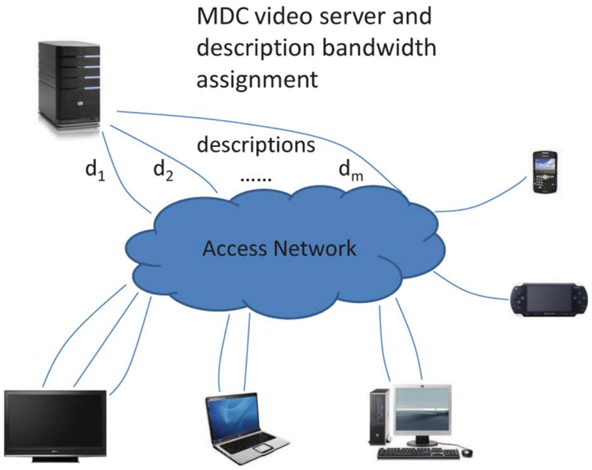

Fig. 1. Video streaming using MDC to heterogeneous users.

To stream video to a large group of users, meeting heterogeneous bandwidth requirements presents a challenging problem. Such bandwidth requirement may differ by orders of magnitude, from hundreds of kbits/s for mobile devices to tens of Mbits/s for high-definition TV. In order to serve all the users, obviously it is neither efficient nor feasible for the server to transcode the stream to each of the user bandwidths. A simple approach is to encode the video into a number of streams of different bitrates, which users join to best match their bandwidth requirements. Given the wide range of bandwidth requirements and limited number of video streams, this approach is clearly not satisfactory, resulting in many receivers getting a stream substantially lower than their bandwidth requirements.

A much better approach is to use multiple description coding (MDC), which encodes the video into multiple independent "descriptions" of different bandwidth [3]–[6]. The descriptions can be arbitrarily combined to best match user's bandwidth requirement. Such approach provides many more options of video bitrates to meet different user requirements, e.g., m descriptions provide up to 2^m different video bitrates [5].[1] A user chooses to receive a set of descriptions, where the sum of their bandwidth best matches the user bandwidth requirement.

We illustrate in Fig. 1 video streaming using MDC to heterogeneous users. The video is encoded into m descriptions with bitrates $d_1, d_2, \dots, d_m$. The users, depending on their access

Manuscript received February 28, 2010; revised September 07, 2010; accepted November 24, 2010. Date of publication December 10, 2010; date of current version March 18, 2011. This work was supported in part by the General Research Fund from the Research Grant Council of the Hong Kong Special Administrative Region, China (611209) and Proof-of-Concept Fund at the HKUST (PCF.005.09/10 and PCFX05-15C00610/11ON). The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Jiangchuan (JC) Liu.

P. Xia and S.-H. G. Chan are with the Department of Computer Science and Engineering, The Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong (e-mail: xiapengye@cse.ust.hk; gchan@cse.ust.hk).

X. Jin is with the Data Infrastructure Team, Facebook, Inc., Palo Alto, CA 94304 USA (e-mail: xjin@fb.com).

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

Digital Object Identifier 10.1109/TMM.2010.2098021

[1] In this paper, we use "user" and "receiver" interchangeably.

1520-9210/$26.00 © 2010 IEEE

bandwidth, get the descriptions that best match their bandwidth requirements so as to maximize their video quality.

In this paper, we study optimal bandwidth assignment for descriptions given heterogeneous bandwidth requirements. We expect an optimal assignment because of the following: if description bandwidths are set too high, the low-bandwidth receivers may not benefit from them (as joining them may exceed their bandwidth), leading to low video quality. On the other hand, if description bandwidths are set too low, those high-bandwidth receivers may not be able to fulfill their bandwidth by joining them, leading again to low video quality. Therefore, we expect optimal description bandwidths to achieve the best overall video quality.

The contributions of our work are as follows.

1) *Problem formulation and complexity analysis:* Given heterogeneous user bandwidth requirements (which can be in terms of a distribution), we formulate an optimization problem for assigning bandwidth to each description so as to maximize overall user satisfaction. The user satisfaction is a function of coding efficiency as well as bandwidth requirement. We prove that the optimization problem is NP-hard.
2) *An exact solution for description number larger than a certain threshold:* Our problem is in general NP-hard. However, when the number of descriptions is larger than or equal to a certain value (e.g., if the minimum and maximum bandwidth requirements are 100 kb/s and 10 Mb/s, respectively, such threshold is seven descriptions), we show that the problem can be solved exactly and efficiently. Our solution takes only $O(m)$ computational time to set m description bandwidths with all user bandwidth requirements fully matched. In other words, maximum overall user satisfaction can be achieved in this case.
3) *An efficient heuristic for smaller description number:* For the case where m is lower than the threshold, we present an efficient heuristic called simulated annealing for MDC bandwidth assignment (SAMBA) to set the description bandwidths. SAMBA is shown to be efficient, and virtually matches the optimum based on exhaustive search. As compared with other simple assignment algorithms, our algorithm can achieve much higher user satisfaction. Using SAMBA, we further show that, if the coding efficiency decreases with the description number, there is an optimal m to achieve maximum user satisfaction. Such m is typically small (in the range of 3–5).

This paper is organized as follows. We review related work in Section II, and describe our problem formulation in Section III. In Section IV, we show an exact assignment method given description number not lower than the threshold, and SAMBA to solve the general problem. Illustrative simulation results and comparisons are presented in Section V. We conclude in Section VI.

II. Related Work

A. Literature Review

MDC has found numerous applications in video streaming [7]–[14]. Much of these previous works on MDC only focus on its error resilient techniques to ensure transmission robustness, and have not considered the assignment of description bandwidth to achieve system performance as investigated in this paper.

Other methods used in multirate video multicast include layered coding (or scalable video coding), where higher layer can be joined only if all the lower layers have been chosen. Performance analysis of layered coding and comparisons with MDC can be found in [15]–[17]. In contrast to layered coding, each description in MDC can be joined or left independently. This flexibility makes MDC a very good candidate to match hetereogeneous video streaming rate requirements. Furthermore, because the descriptions are not coupled as tightly as in layered coding, MDC can achieve better error containment in case packets are lost. There has been work on optimal bandwidth assignment for layered coding [18]. However, the techniques used for layered coding cannot be directly applied to MDC because the complexity MDC optimization is much higher due to the combinatorial nature of the problem. MDC has higher coding inefficiency as compared with layered coding [8], [19]–[21]. Seldom has there been work on how to optimally assign description bandwidth with such efficiency consideration. We study this issue here.

The work in [22]–[24] has addressed optimal bandwidth assignment problem for MDC. We differ here in several major ways. Firstly, our paper provides more general formulation with coding efficiency consideration. Furthermore, we show that when the description number is no less than a certain threshold, there is a simple and efficient algorithm to achieve exact optimum which matches all the bandwidth requirements in the network. We also show that there is an optimal description number to achieve the best user satisfaction, which has never been studied in the literature before. Our approach based on simulated annealing is efficient and achieves performance close to exhaustive search.

B. Review of Simulated Annealing

Because our efficient heuristic SAMBA is based on simulated annealing, we review its principles here. Simulated annealing was first proposed by Kirkpatrick *et al.* in 1983 as a framework to find approximate solution for difficult combinatorial optimization problems [25]. It takes the concept from statistical mechanics and is inspired by the behavior of the physical system in the annealing process.[2]

Given a combinatorial problem, we try to find the best solution of lowest cost which is defined by a cost function. Conventional iterative improvement heuristics iteratively find a solution of lower cost by making small local alternations and stop at a local optimum. In contrast, simulated annealing has a better chance to approach global optimum. The key idea is that in each iteration, the acceptance of a solution of higher cost is probabilistic. In other words, simulated annealing occasionally allows an "uphill" movement instead of always searching "downhill" to avoid being trapped in the local optima. The probability of accepting a higher cost solution is controlled

[2] Annealing is a process of heating and slowly cooling down to toughen a subject and reduce its brittleness.

by the temperature parameter. The higher the temperature, the more probable the acceptance of a higher cost solution. The temperature parameter is initially high and then slowly lowered down, which is analogous to the temperature in the real annealing process. As stated in [25], four ingredients are needed to apply simulated annealing to a combinatorial problem:

- definition of a state (or configuration) of the system, which is usually a point in the search space;
- an energy (or cost) function that can be evaluated at each state, which is usually equivalent to the cost function in the combinatorial problem;
- a transition function which picks a neighbor state, evaluates its energy, and decides whether the system moves to the state based on the probability;
- an annealing schedule to control the initialization and lowering down of the temperature parameter.

Given the four ingredients, a generic simulated annealing algorithm can be stated as follows.

- *Step 1)* Set the initial state and the initial temperature.
- *Step 2)* Make a transition from the current state to a neighbor state.
- *Step 3)* Repeat Step 2 for a number of times, record down the top state of lowest energy.
- *Step 4)* Lower down the temperature according to the annealing schedule. If the end of the schedule is reached, return the top state and its energy. Otherwise, repeat Steps 2 and 3 by using the top state from previous temperature as the starting state.

The framework provides freedom for the design of the transition function and the annealing schedule. The transition function contains two major steps. On the one hand, in each iteration, the function should randomly pick a state from the neighborhood. The definition of the neighborhood depends on the given problem. For instance, in the example of the traveling salesmen problem in [25], a state is defined as a permutation list of the cities and its neighbor state is defined as another list which can be generated by reversing any section of the permutation list. For another instance, in image restoration problem in [26], the neighborhood is defined by the neighborhood radius which is related to the temperature. On the other hand, given a neighbor state, the transition function should decide the transition probability. In [25], the author suggests that when the neighbor state has lower energy, the probability is one; otherwise, the probability is proportional to the Boltzmann factor of the energy difference. The detail annealing schedule (including how to initialize and to lower down the temperature) depends on the nature of the given problem and can be developed by trial and error [25]. In the example of partition problem in [25], the temperature is cooled down exponentially.

III. Problem Formulation

Consider a video stream to be accessed by a large pool of users with heterogeneous bandwidth requirements. In Fig. 2, we show the optimization being done at the server. The server encodes the stream into multiple descriptions, given the description number m. User j has bandwidth requirement c_j and a certain importance according to a weight value w_j. Users employ a greedy approach in joining the descriptions to maximize their video quality, i.e., they join the descriptions so that the total bandwidth best matches the bandwidth requirement without overflowing.[3] We show some of the important symbols used in the paper in Table I.

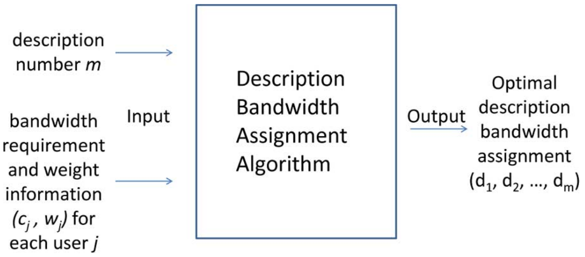
description number m
bandwidth requirement and weight information (c_j, w_j) for each user j
Input
Description Bandwidth Assignment Algorithm
Output
Optimal description bandwidth assignment (d_1, d_2, ..., d_m)

Fig. 2. Optimization model for MDC bandwidth assignment.

TABLE I
Major Symbols Used in the Paper and Their Explanations

Symbol	Explanation
d_i	i^{th} description bandwidth
m	description number
n	number of users
h	heterogeneity factor
c_j	bandwidth requirement for user j
v_j	total joined video bandwidth
r_j	bandwidth matching factor, defined as the ratio of v_j and c_j
w_j	weight for user j, which represents the user importance
α_m	coding efficiency factor given m descriptions
β	base factor used to model coding efficiency factor
$\vec{d}$	description bandwidth assignment vector
S_{ind}	individual satisfaction given description bandwidths and user bandwidth requirement
S	overall network satisfaction given description number and bandwidth assignment
S^*	optimal overall satisfaction
f	individual satisfaction function, which is a function of r_j and α_m
K	matrix when K_{ij}=1 if and only if user i joins description j

Denote the bandwidth of description i by d_i. Then $\vec{d}$ is m-dimensional vector (sorted in the increasing order) and represents a particular bandwidth assignment for the descriptions. The total joined video bandwidth v_j is the sum of received description bandwidths. Clearly we need

$$v_j \leq c_j. \tag{1}$$

Let $K_{ij} \in \{0, 1\}$ be a binary number with 1 indicating that user j chooses description i. We have

$$v_j = \sum_{i=1}^{m} K_{ij} d_i. \tag{2}$$

We consider that bandwidth is normalized to some unit (say, 50 kb/s), and hence c_j and d_i are integral. The heterogeneity

[3] Note that the bandwidth requirement may be represented by a distribution. If this is the case, then there is a certain fraction of users with bandwidth requirement c_j according to the distribution, and w_j is the importance of each of the users in the group. The remaining study extends naturally and straightforwardly to this case.

factor h is defined as the difference between minimum and maximum user bandwidth requirement, i.e.,

$$h = \max_j c_j - \min_j c_j + 1. \tag{3}$$

Define r_j the bandwidth matching factor given by the ratio of v_j and c_j, i.e.,

$$r_j = \frac{v_j}{c_j}. \tag{4}$$

Define $\alpha_m \in (0, 1]$ as the coding efficiency factor given m descriptions, which decreases with m. We model user individual satisfaction as a monotonically increasing function f in terms of $r_j \alpha_m$. The individual satisfaction of user j is hence given by

$$S_{ind}(\vec{d}, c_j) = f(\alpha_m r_j). \tag{5}$$

Let n be the number of users in the network. The overall network satisfaction is hence given by

$$S(\vec{d}) = \frac{\sum_{j=1}^{n} w_j S_{ind}(\vec{d}, c_j)}{\sum_{j=1}^{n} w_j}. \tag{6}$$

Our objective is then to find optimal bandwidth assignment $\vec{d}^*$ so as to maximize (6) subject to (1), (2), (4), and (5), i.e.,

$$S^* = S(\vec{d}^*) = \max_{\vec{d}} S(\vec{d}). \tag{7}$$

The problem is NP-hard (shown in the Appendix, by finding a polynomial reduction from the subset sum problem).

IV. Algorithms for Optimal Assignment of Description Bandwidths

In Section IV-A, we present the threshold value for description number. We show that when the description number is no less than the threshold, there is a simple and efficient assignment algorithm to match all the user bandwidth requirements. In Section IV-B, we present an efficient heuristic SAMBA to assign description bandwidth for the more general case.

A. Threshold and the Exact Solution

Consider that user bandwidth requirement ranges in $[a, b]$, where a and b are the maximum and minimum user bandwidth requirement, i.e., $a = \min_j c_j$ and $b = \max_j c_j$.

Let us first consider the simple case where a is equal to one (i.e., equal to the basic unit). All the values in $[a, b]$ can be converted to a binary number by changing base to 2. The number of binary digits for a particular value is bounded by the number of digits of b in binary form, which is clearly $\lfloor \log_2 b \rfloor + 1$.

A binary number can be regarded as a linear combination of 2's powers with coefficients either 0 or 1. For example, the binary form of 25 is 11001. If the description bandwidth is assigned to be a power of 2 (i.e., $1, 2, 2^2, \ldots, 2^{m-1}$), then the binary form of the bandwidth requirement represents exactly the joining choice, with coefficient 1 to join the corresponding description and 0 otherwise. As in the example above, the user with bandwidth 25 units will choose to join descriptions with bandwidth 16, 8, and 1 units. The maximum number of binary digits indicates the description number, which is $\lfloor \log_2 b \rfloor + 1$.

From above, it is clear that if $m \geq \lfloor \log_2 b \rfloor + 1$, all the bandwidth requirements can be fully matched, because their values can be expressed as 0–1 linear combination of the description bandwidths.

Now consider a to be any integer larger than one. In this case, bandwidth requirement can be expressed as $(a-1) + x$, where $x \in [1, b - a + 1]$. By the previous argument, x can be written in binary form with at most $\lfloor \log_2 (b - a + 1) \rfloor + 1$ digits. Then any bandwidth requirement can be expressed as a linear combination of a description bandwidth $(a-1)$ and the binary expression for x. According to the definition of heterogeneity factor in (3), we have $h = b - a + 1$. Therefore, the threshold value for the description number in terms of h is

$$\lfloor \log_2 h \rfloor + 2. \tag{8}$$

If m is no less than this value, the description bandwidths are hence $\{(a-1), 2^0, 2^1, \ldots, 2^{\lfloor \log_2 h \rfloor}\}$. With this assignment, all the bandwidth requirements can be fully matched.

It simply takes $O(m)$ computations to decide the bandwidths for the descriptions, given m is no less than the threshold.

B. SAMBA

In this section, we present an efficient heuristic SAMBA to solve the general problem when description number m is no larger than the threshold given by (8).

If m is less than the threshold, the problem is to search in an m-dimensional integer space for the optimal description bandwidths. The search space is discrete and finite, because each description can only take integral bandwidth no larger than the maximum bandwidth requirement in the network. This condition makes it feasible to adopt simulated annealing algorithm (Section II-B) to solve the problem.

In SAMBA, a state is defined as a vector $\vec{d}$ of description bandwidths (sorted in the increasing order). Each state is associated with an "internal energy", which is defined to be the negative of the satisfaction value (as in simulated annealing, we seek to *minimize* an objective function). SAMBA starts with an initial state and iteratively transits to other state seeking for lower internal energy (and hence higher user satisfaction).

Each state has a neighborhood given by a radius r. By saying state $\vec{d_1}$ is a neighbor of state $\vec{d_2}$, we mean $\|\vec{d_1} - \vec{d_2}\| < r$ (we use l_1-norm in the simulation). At each iteration, SAMBA randomly picks a neighbor of the current state as the target state, and decides whether or not to make the transition according to a transition probability p.

We further define a temperature T, which exponentially decreases as the algorithm iterates. Let i be the current iteration number and K be the total number of iterations. We have

$$T = e^{i/K} - e^{-1}. \tag{9}$$

The higher T is, the larger is the neighborhood radius r. We define r as

$$r = \max\left\{2, \frac{hT}{1 - e^{-1}}\right\}. \tag{10}$$

We then define transition probability as follows. Denote the current state by $\vec{d}$ and target state by $\vec{d_t}$. The energy difference between the two states is given by the difference of their satisfaction value, i.e., $S(\vec{d})-S(\vec{d_t})$. If $S(\vec{d})-S(\vec{d_t}) < 0$, the transition probability p is defined as 1. Otherwise, p is a decreasing function of $(S(\vec{d})-S(\vec{d_t}))/T$ with initial value close to 1. Given T, the lower the satisfaction of the target state is, the smaller is the transition probability. Further, with large T, any target state has transition probability near 1. Therefore, we have

$$p(\vec{d},\vec{d_t}) = \begin{cases} 1, & \text{if } S(\vec{d_t}) > S(\vec{d}) \\ e^{S(\vec{d_t})-S(\vec{d})/T}, & \text{otherwise.} \end{cases} \quad (11)$$

This definition of p is equivalent to the definition of transition probability in [25] which uses Boltzmann factor.

At the early iterations when the temperature is high, SAMBA picks the target state from a large neighborhood. Because the transition probability to any picked state is high, the algorithm randomly moves among the states. As the algorithm iterates, the temperature gets lower. SAMBA picks the target state from a smaller neighborhood and the transition probability to a state of lower satisfaction decreases. In other words, the algorithm gradually settles to a neighborhood with locally-optimal satisfaction. By running SAMBA with different initial states, we have great chance to find the global optimum. The whole algorithm can hence be summarized in the following steps.

- *Step 0*) For the first iteration, set the initial temperature value and the initial state $\vec{d_0}$. Find out initial satisfaction S_0. Then set the highest satisfaction $S_{\max} = S_0$ and its associated state $\vec{d}_{\max} = \vec{d_0}$.
- *Step 1*) Update the temperature value.
- *Step 2*) Find a target state $\vec{d_t}$ in the neighborhood and evaluate its satisfaction S. If $S > S_{\max}$, assign $\vec{d_t}$ and S to $\vec{d}_{\max}$ and $S_{\max}$, respectively.
- *Step 3*) Make the transition decision according to the transition probability.
- *Step 4*) Repeat Steps 2 to 3 for a number C times.
- *Step 5*) Set current state to $\vec{d}_{\max}$ and repeat Steps 1 to 4 for a number K iterations. Return $S_{\max}$ and $\vec{d}_{\max}$.

The computational complexity of our algorithm is as follows. According to Steps 4 and 5, Steps 1 to 3 are repeated for KC times. The first and the third steps take constant computations while Step 2 has to calculate the overall satisfaction for the target state. According to (6), S is a weighted sum of individual satisfactions S_{ind}. Given $\vec{d}$ and c_j, evaluating each S_{ind} becomes an integer Knapsack problem and takes at most $O(2^m)$ time. Therefore, given n users, Step 2 takes $O(n2^m)$ time. Thus, the whole algorithm runs in $O(KCn2^m)$ time. In contrast, the exhaustive search has to evaluate S for all the states in order to decide the optimal overall satisfaction. Each state ($\vec{d}$ in sorted order) is a combination of m integers in $[1, \max_j c_j]$. The number of such combinations is lower bounded by $(\max_j c_j)^m/m!$. Given (3), we can further bound the number of states by

$$\frac{(\max_j c_j)^m}{m!} \geq \frac{h^m}{m!} = \prod_{i=1}^{m} \frac{h}{i}. \quad (12)$$

SAMBA evaluates S for KC times, which is usually much smaller than $\prod_{i=1}^{m}(h/i)$. This is why SAMBA is much more efficient than exhaustive search.

So far, we are given the description number m. There is in fact an optimal m to achieve maximum user satisfaction among all m's. This is expected as α_m is a decreasing function in m (due to decreasing coding efficiency) while r_j is an increasing function in m (due to better bandwidth matching). Because all the bandwidth requirements are fully matched if m is larger than the threshold as given in (5), the optimal m to achieve maximum user satisfaction is bounded by the threshold. To search for the optimal description number, we can run SAMBA for increasing m up to the threshold in the simulation.

V. Illustrative Simulation Results

In this section, we present illustrative simulation results to show the efficiency of our algorithm. In Section V-A, we describe simulation environment. In Section V-B, we compare SAMBA with other bandwidth assignment schemes and examine the influence of some important factors.

A. Simulation Environment and Parameters

In our simulation, we have compared SAMBA with exhaustive search, which always achieves the optimum. Besides, we have also compared SAMBA with other simple bandwidth assignment schemes, which include uniform assignment (in which all the descriptions have the same bandwidth), linear assignment (in which the description bandwidth is linearly increased), exponential assignment (in which the description bandwidth is exponentially increased), and random assignment (which randomly assigns bandwidth for each description).

We consider a simple user individual satisfaction function given by $f(\alpha_m r_j) = (\alpha_m r_j)^k$, where $k \in R^+$. The function is reasonable as it is strictly increasing in $[0, 1]$ and the minimum and maximum are 0 and 1, respectively. For concreteness, we consider coding efficiency as $\alpha_m = \beta^{m-1}$, where $\beta < 1$ (the value of β depends on the underlying MDC techniques used).

We consider that the number of users of bandwidth requirement c is proportional to some distribution given by $N(c)$. We use a simple truncated normal curve as the bandwidth distribution, i.e., $c \sim N_T(\mu, \sigma^2)$. Fig. 3 show a truncated normal $c \sim N_T(10, 5)$.

Each user has its "importance" in terms of a weight $w(c)$, which is a function of bandwidth requirement. According to (6), simulation on $w(c)$ with uniform bandwidth requirement distribution is equivalent to simulation on $N(c)$ with constant weight. Therefore, we only focus on the influence of $N(c)$ and expect the similar result for $w(c)$.

Unless otherwise stated, we use baseline parameters $k = 2$, $\beta = 1$, $m = 3$, and $v_j \in [1, 100]$.

B. Illustrative Results

Fig. 4 plots the overall satisfaction S versus heterogeneity factor h given different bandwidth assignment schemes. The satisfaction S is decreasing with h. This is because it becomes

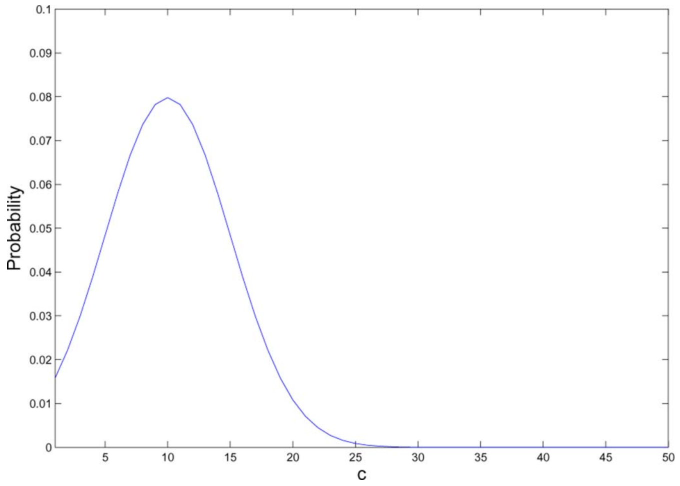

Fig. 3. Example bandwidth requirement distribution $c \sim N_T(10, 5)$.

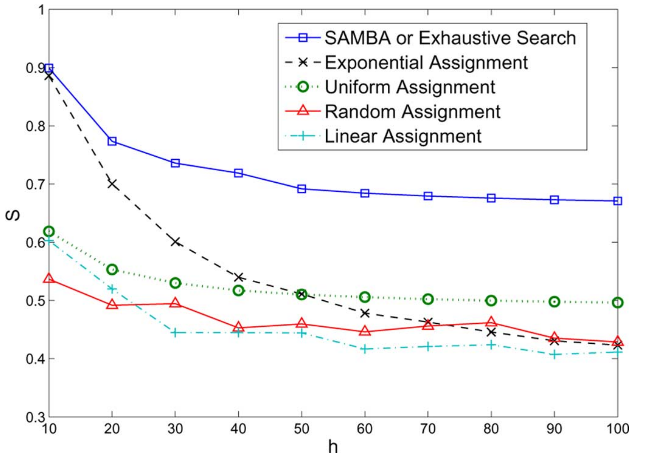

Fig. 4. Overall satisfaction (S) versus heterogeneity factor h given different schemes.

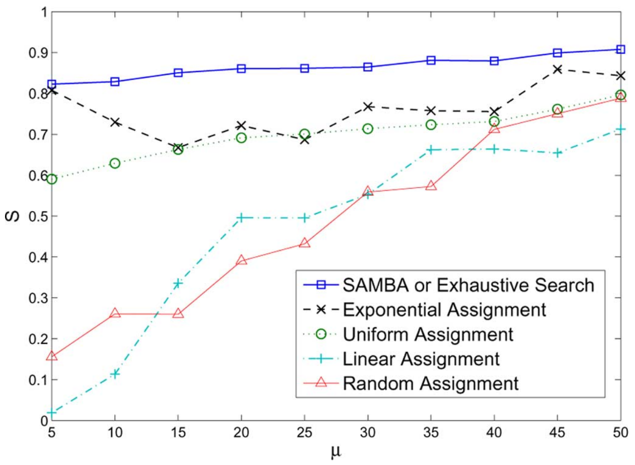

Fig. 5. Overall satisfaction (S) versus mean value (μ) given different schemes.

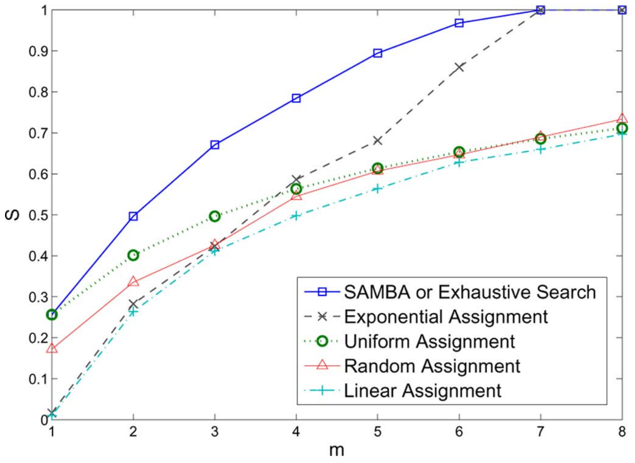

Fig. 6. Overall satisfaction (S) versus description number (m) given different schemes.

more difficult to match the bandwidth requirements if heterogeneity factor gets higher. In the graph, for each h, overall satisfaction given by SAMBA overlaps with that given by exhaustive search, and is much better than those given by the other schemes. SAMBA performs virtually the same as exhaustive search. Exponential assignment performs better than uniform assignment, random assignment, and linear assignment when h is not large. Besides, random assignment and linear assignment have the similar performance and uniform assignment has the better performance than them.

Fig. 5 plots the overall satisfaction S versus the mean value μ given different bandwidth assignment schemes. If a scheme is robust to the variation of bandwidth requirement distribution, its overall satisfaction should not change much with different μ. From the graph, S given by SAMBA overlaps with that given by exhaustive search and is not affected by the changes of μ. However, S given by the other schemes is lower when μ is small. The comparison shows SAMBA performs as well as exhaustive search and is robust when the bandwidth requirement distribution changes. Besides, exponential assignment performs better than uniform assignment and much better than linear assignment and random assignment. Linear assignment and random assignment have the similar performance.

Fig. 6 plots the overall satisfaction S versus number of descriptions m given different bandwidth assignment schemes. The overall satisfaction S increases with m. This is because more descriptions can provide more options of bitrates to meet heterogeneous user bandwidth requirements. For each m, SAMBA performs as well as exhaustive search[4] and much better than the other schemes. The satisfaction of SAMBA finally settles to a value, because all the bandwidth requirements are fully matched after m reaches the threshold. Exponential assignment achieves the same performance after m reaches the threshold. This is because, in this case, the exact assignment in Section IV-A is optimal and its descriptions have exponentially increasing bandwidths. Uniform assignment, linear assignment, and random assignment have the similar performance.

[4] Due to high computational complexity, it becomes infeasible to find overall satisfaction using exhaustive search when m is large. Therefore, we only plot S given by exhaustive search for $m \leq 3$, which overlaps with S given by SAMBA.

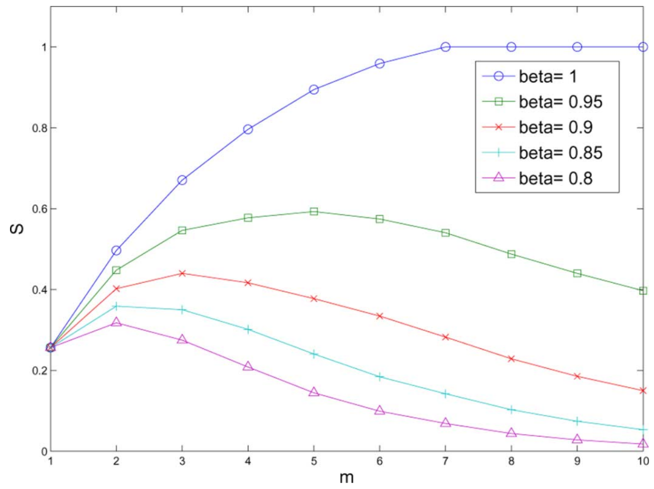

Fig. 7. Overall satisfaction (S) versus description number (m) given β.

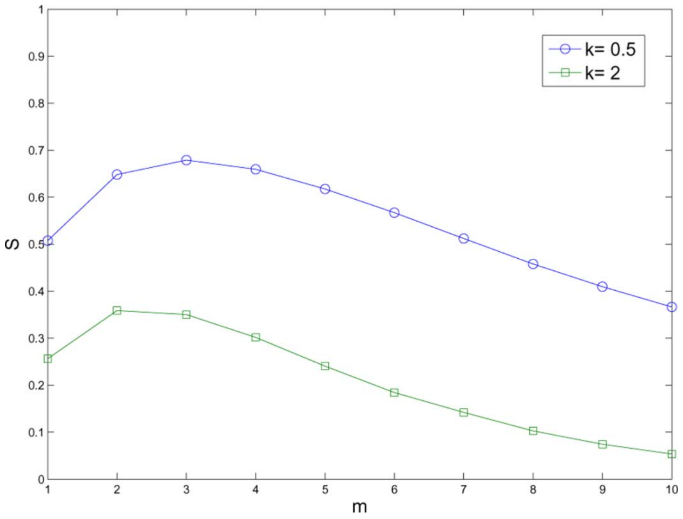

Fig. 8. Overall satisfaction (S) versus description number (m) given k.

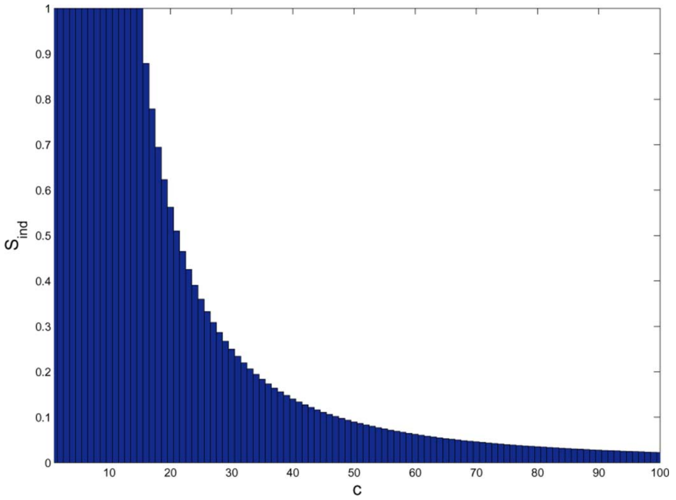

Fig. 9. Individual satisfaction (S) versus bandwidth requirement with $c \sim N_T(0, 10)$.

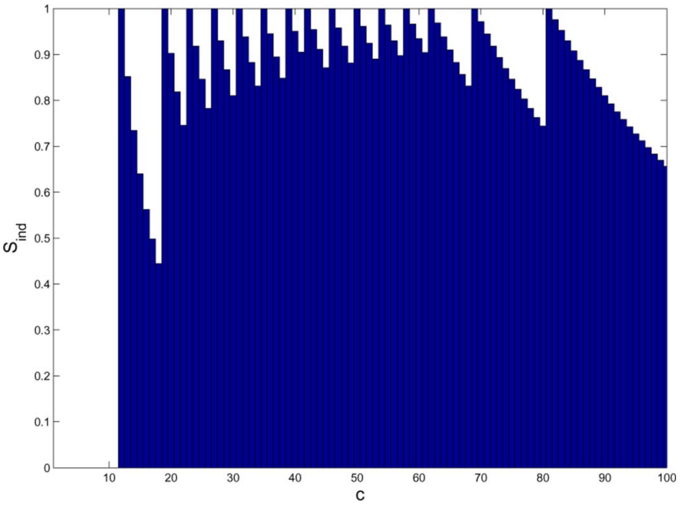

Fig. 10. Individual satisfaction (S) versus bandwidth requirement with $c \sim N_T(50, 10)$.

We next examine the influence of coding efficiency. Fig. 7 plots the overall satisfaction S versus number of descriptions m given different β. Given decreasing coding efficiency (β less than 1), the overall satisfaction first increases with m and then decreases with m. The reason is as follows. When m is small, satisfaction is low because the bandwidth requirements are badly matched (i.e., r_j is small). When m is large, satisfaction is also low because the coding efficiency is low (i.e., α_m is small). Therefore, we expect the optimal m when the effect of decreasing coding efficiency and the effect of better bandwidth matching balances with each other. From the figure, we observe that the optimal m becomes smaller when coding efficiency decreases faster (i.e., β is smaller). In the MDC implementation, m should be chosen according to β.

We then examine the influence of k (the coefficient to model satisfaction function f). Fig. 8 plots the overall satisfaction S versus description number m given different k. S first increases with m then decreases with m. This is due to decreasing coding efficiency (with $\beta = 0.85$). And the optimal m is depending on the choice of k. The reason is as follows. Function f has different convexity with different k. When f is convex ($k = 2$), the satisfaction increases more rapidly when the bandwidth matching factor r_j approaches 1. In this case, users emphasize more on high-quality service. In the other case, when f is concave ($k = 0.5$), the satisfaction increases more rapidly when r_j is small, which implies users emphasize more on the availability of the content. As a result, optimal m may change given different k. For each m, satisfaction is lower if $k = 2$, because $x^2 \leq x^{0.5}$ for any $x \in [0, 1]$.

We finally examine the influence of bandwidth requirement distribution. Figs. 9–11 plot the individual satisfaction S_{ind} values for each bandwidth requirement c given different bandwidth requirement distributions $c \sim N_T(0, 10)$, $c \sim N_T(50, 10)$, and $c \sim N_T(100, 10)$, respectively. From the figures, users with bandwidth requirements around μ have higher individual satisfaction. The reason is as follows. The number of users $N(c)$ is larger, if c is closer to μ. According to the (6), if $N(c)$ is larger, $S_{ind}(\vec{d}, c)$ has greater influence on S. Therefore, SAMBA satisfies users of bandwidth c with higher

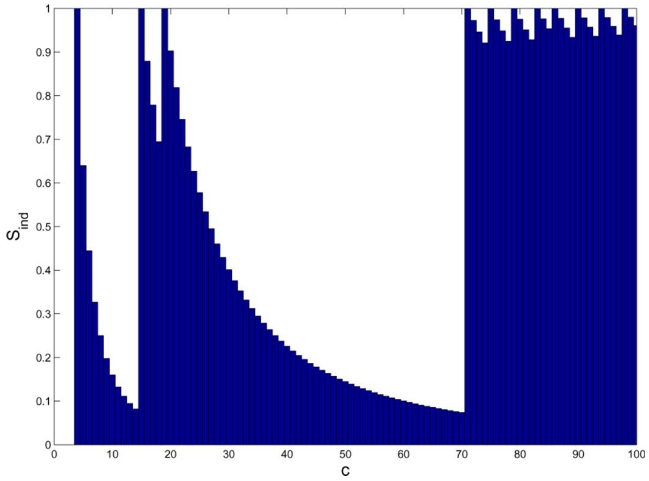

Fig. 11. Individual satisfaction (S) versus bandwidth requirement with $c \sim N_T(100, 10)$.

TABLE II
DESCRIPTION BANDWIDTHS ASSIGNED BY SAMBA

Distribution	d_1	d_2	d_3	d_4	S
$N_T(0, 10)$	1	2	4	8	0.9460
$N_T(50, 10)$	12	19	23	27	0.9362
$N_T(100, 10)$	4	7	15	72	0.9670

precedence to maximize S. Further, we expect the influence of $w(c)$ is similar (the users of bandwidth requirement c get higher individual satisfaction if their weight $w(c)$ is larger).

Table II shows the description bandwidths assigned by SAMBA and the corresponding overall satisfaction, given the above three different bandwidth requirement distributions. Note that most of the users are with bandwidth requirement around the mean of the corresponding distribution. It is easy to see that the assignment as shown by each row significantly reduces the bandwidth mismatch for these users.

VI. CONCLUSION

In this paper, we study how to optimally assign description bandwidth for MDC for video streaming to large group, so that their heterogeneous bandwidth requirements can be best satisfied. We formulate the problem as an optimization problem and propose algorithms to address it.

We have proved that the formulated optimization problem is NP-hard, by finding a polynomial reduction to the problem from the subset sum problem. We show that when the description number is no smaller than a certain threshold, simple and efficient assignment algorithm of run-time complexity of $O(m)$ can fully match all the bandwidth requirements. For the general case when m is less than the threshold, we have proposed and studied the heuristic SAMBA, which uses simulated annealing to efficiently find the optimal description bandwidth assignment. Furthermore, there exists an optimal choice for description number to achieve maximum user satisfaction.

Simulation results have shown that SAMBA achieves much better user satisfaction than other assignment methods and closely matches the optimum based on exhaustive search. With the consideration of coding efficiency, we show that indeed there is an optimal description number to achieve maximum user satisfaction.

APPENDIX

In this section, we prove our problem is NP-hard. We find a polynomial reduction from the subset sum problem L_c to our problem L. In order to have $L_c \leq_P L$, we prove any input of L_c can be transformed into input of L in polynomial time. Also the output of L is equivalent to that of L_c and can be transformed back in polynomial time as well.

The input of L_c is $(x_1, x_2, \ldots, x_m, t)$, with all x_i and t positive integers. The output is a "yes" or "no" to decide whether t is sum of a subset of $\{x_i\}$. We assume $x_i \neq 0$, because after excluding zeros the problem is still equivalent to the original one. We construct the input for L in the form of $(x_1, x_2, \ldots, x_m, \sum_i x_i, t, m)$, where the first $m+2$ elements are user bandwidth requirements and the last one is the description number. We further set $\alpha(m) = 1$ and let all users have the same weight. until this point, we have transformed the input of L_c to that of L, which obviously takes polynomial time.

The output of L is $(d_1, d_2, \ldots, d_m, S)$, where d_i is description bandwidth and S is the optimal overall satisfaction. According to (6), maximum of S is 1 when all the bandwidth requirements are fully matched i.e., $S \leq 1$. If t is a subset sum in L_c, t can be expressed as a linear combination of x_i with zero or one coefficients. Then if we let descriptions have bandwidths $(x_1, x_2, \ldots, x_m)$, all the bandwidth requirements in L are fully matched. We thus have $S = 1$. In other words, $(x_1, x_2, \ldots, x_m, 1)$ must be one of the solutions of L. If we show it is also the only solution, we can prove the equivalence between the outputs of L_c and L, i.e., "t is subset sum in L_c" $\Leftrightarrow$ "L has output $(x_1, x_2, \ldots, x_m, 1)$".

Now let $d = (d_1, d_2, \ldots, d_m)^T$ be the optimal descriptions bandwidth assignment for L. There exists an $(m+2)$ by m matrix A with each element a_{ij} be zero or one, such that

$$Ad = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \\ \sum x_i \\ t \end{pmatrix}. \tag{13}$$

By adding the first m rows together, we have the following equation:

$$\left(\sum_{i=1}^{m} a_{i1}, \sum_{i=1}^{m} a_{i2}, \ldots, \sum_{i=1}^{m} a_{im} \right) \cdot \begin{pmatrix} d_1 \\ d_2 \\ \vdots \\ d_m \end{pmatrix} = \sum_{i=1}^{m} x_i. \tag{14}$$

We claim that none of the $\sum a_{ij}$ equals zero. Because if it is true, we must have a column of zero in A and a d_j without any user joining it. In this case, we can simply remove the column and the description. Then we have

$$\sum_{i=1}^{m} a_{ij} \geq 1 \tag{15}$$

for all j. Using the $(m+1)$th row of matrix A, we have

$$\sum_{i=1}^{m} x_i \tag{16}$$

$$= (a_{(m+1)1}, a_{(m+1)2}, \ldots, a_{(m+1)m}) \cdot \begin{pmatrix} d_1 \\ d_2 \\ \vdots \\ d_m \end{pmatrix} \tag{17}$$

$$\leq (1, 1, \ldots, 1) \cdot \begin{pmatrix} d_1 \\ d_2 \\ \vdots \\ d_m \end{pmatrix} \tag{18}$$

$$\leq \left(\sum_{i=1}^{m} a_{i1}, \sum_{i=1}^{m} a_{i2}, \ldots, \sum_{i=1}^{m} a_{im} \right) \cdot \begin{pmatrix} d_1 \\ d_2 \\ \vdots \\ d_m \end{pmatrix} \tag{19}$$

$$= \sum_{i=1}^{m} x_i \tag{20}$$

which implies, after some manipulations

$$a_{(m+1)j} = 1 \tag{21}$$

and

$$\sum_{i=1}^{m} a_{ij} = 1, \forall\, j. \tag{22}$$

Since $\sum_{i=1}^{m} a_{ij}$ is the sum of first m rows for column j, if it equals one, there is exactly one entry a_{ij} in this column with value one. Because there are m such columns (or less than m if zero columns are removed), there are at most m entries of value one in the top m rows of matrix A. Now if there is one row containing two entries with value one, there must be at least one row with all zero entries. It is equivalent to say that one of x_i is zero, which is contradicting to our assumption. Therefore, each row must contain exactly one entry with value one. (If the number of columns is less than m, there must be one row with all zero entries leading to the same contradiction.) By looking at the top m by m matrix, we can conclude that

$$(d_1, d_2, \ldots, d_m) = (x_{i_1}, x_{i_2}, \ldots, x_{i_m}) \tag{23}$$

where the right-hand side is a permutation of x_i. In other words, the set x_i is the only solution which fully matches all the bandwidth requirements. Therefore, we have proved the equivalence between output of L and L_c. The transformation between the outputs is obviously in polynomial time.

Therefore, we conclude that there exists polynomial reduction from L_c to L and our problem L is NP-hard.

References

[1] Y. T. Yun, J.-G. L. Jian-Guang, M. Z. Meng, S.-Q. Y. Shi-Qiang, and Q. Z. Qian, "Deploying P2P networks for large-scale live video-streaming service [peer-to-peer multimedia streaming]," *IEEE Commun. Mag.*, vol. 45, no. 6, pp. 100–106, Jun. 2007.

[2] Y. F. Chen, Y. Huang, R. Jana, H. Jiang, M. Rabinovich, B. Wei, and Z. Xiao, "When is P2P technology beneficial for IPTV services," in *Proc. ACM NOSSDAV*, Jun. 2007.

[3] V. N. Padmanabhan, H. J. Wang, P. A. Chou, and K. Sripanidkulchai, "Distributing streaming media content using cooperative networking," in *Proc. 12th Int. Workshop Network and Operating Systems Support for Digital Audio and Video*, New York, May 2002, pp. 177–186. [Online]. Available: http://doi.acm.org/10.1145/507670.507695.

[4] M. Castro, P. Druschel, A. M. Kermarrec, A. Nandi, A. Rowstron, and A. Singh, "Splitstream: High-bandwidth multicast in cooperative environments," in *Proc. ACM SOSP'03*, Oct. 2003, pp. 298–313.

[5] B. Li and J. Liu, "Multirate video multicast over the internet: An overview," *IEEE Netw.*, vol. 17, no. 1, pp. 24–29, Jan./Feb. 2003.

[6] M.-T. Lu, J.-C. Wu, K.-J. Peng, P. Huang, J. Yao, and H. Chen, "Design and evaluation of a P2P IPTV system for heterogeneous networks," *IEEE Trans. Multimedia*, vol. 9, no. 8, pp. 1568–1579, Dec. 2007.

[7] E. Akyol, A. Tekalp, and M. Civanlar, "A flexible multiple description coding framework for adaptive peer-to-peer video streaming," *IEEE J. Select. Topics Signal Process.*, vol. 1, no. 2, pp. 231–245, Aug. 2007.

[8] T. Tillo and G. Olmo, "Data-dependent pre-and postprocessing multiple description coding of images," *IEEE Trans. Image Process.*, vol. 16, no. 5, pp. 1269–1280, May 2007.

[9] Y. Xu and C. Zhu, "Joint multiple description coding and network coding for wireless image multicast," in *Proc. 5th Int. Conf. Image and Graphics (ICIG '09)*, Sep. 2009, pp. 819–823.

[10] O. Badarneh, M. Kadoch, and A. Elhakeem, "Video multicast based multiple description coding and multi-paths in wireless ad HOC networks," in *Proc. Canadian Conf. Electrical and Computer Engineering (CCECE '09)*, May 2009, pp. 604–609.

[11] M. Ma, O. Au, L. Guo, S.-H. Chan, and P. Wong, "Error concealment for frame losses in MDC," *IEEE Trans. Multimedia*, vol. 10, no. 8, pp. 1638–1647, Dec. 2008.

[12] M. Firooz, K. Ronasi, M. Pakravan, and A. Avanaki, "Wavelet-based unbalanced un-equivalent multiple description coding for P2P networks," in *Proc. IEEE Int. Conf. Telecommunications and Malaysia Int. Conf. Communications (ICT-MICC 2007)*, May 2007, pp. 242–247.

[13] O. Crave, B. P. Popescu, and C. Guillemot, "Robust video coding based on multiple description scalar quantization with side information," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 20, no. 6, pp. 769–779, Jun. 2010.

[14] S. Milani and G. Calvagno, "Multiple description distributed video coding using redundant slices and lossy syndromes," *IEEE Signal Process. Lett.*, vol. 17, no. 1, pp. 51–54, Jan. 2010.

[15] M. Wien, H. Schwarz, and T. Oelbaum, "Performance analysis of SVC," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 9, pp. 1194–1203, Sep. 2007.

[16] Y. Zhou and W.-Y. Chan, "Performance comparison of layered coding and multiple description coding in packet networks," in *Proc. IEEE Global Telecommunications Conf. (GLOBECOM '05)*, Dec. 2005, vol. 4.

[17] Y.-C. Lee, J. Kim, Y. Altunbasak, and R. Mersereau, "Performance comparisons of layered and multiple description coded video streaming over error-prone networks," in *Proc. IEEE Int. Conf. Communications (ICC '03)*, May 2003, vol. 1, pp. 35–39.

[18] J. Liu, B. Li, and Y.-Q. Zhang, "A hybrid adaptation protocol for tcp-friendly layered multicast and its optimal rate allocation," in *Proc. IEEE 21st Annu. Joint Conf. IEEE Computer and Communications Societies (INFOCOM 2002)*, Jun. 2002, vol. 3, pp. 1520–1529.

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

[20] H. S. Witsenhausen and A. D. Wyner, "Source coding for multiple descriptions ii: A binary source," *Bell Syst. Tech. J.*, vol. 60, no. 10, pp. 2281–2292, Dec. 1981.

[21] R. Venkataramani, G. Kramer, and V. Goyal, "Multiple description coding with many channels," *IEEE Trans. Inf. Theory*, vol. 49, no. 9, pp. 2106–2114, Sep. 2003.

[22] T. Abanoz and A. Tekalp, "Optimization of encoding configuration in scalable multiple description coding for rate-adaptive P2P video multicasting," in *Proc. 16th IEEE Int. Conf. Image Processing (ICIP)*, Nov. 2009, pp. 3741–3744.

[23] B. Li, J. Xu, J. Liu, B. Li, and X. Cao, "Bandwidth provisioning for multiple description coding based video distribution," in *Proc. 5th Int. Symp. Wireless Personal Multimedia Communications*, Oct. 2002, vol. 2, pp. 802–806.

[24] P. Xia, X. Jin, and S.-H. Chan, "Optimal bandwidth assignment for multiple description coding in media streaming," in *Proc. 6th IEEE Consumer Communications and Networking Conf. (CCNC 2009)*, Jan. 2009, pp. 1–5.

[25] S. Kirkpatrick, C. D. Gelatt Jr., and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, pp. 671–680, May 1983.

[26] S. Geman and D. Geman, "Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-6, no. 6, pp. 721–741, Nov. 1984.

Pengye Xia received the B.Sc. degree in physics and computer science and the M.Phil. degree in computer science and engineering from The Hong Kong University of Science and Technology (HKUST), Kowloon, Hong Kong, in 2009 and 2010, respectively.

He is currently a Research Assistant in the Multimedia Technology Research Center, HKUST. His research interest includes multimedia networking, network optimization, and distributed algorithms.

S.-H. Gary Chan (S'89–M'98–SM'03) received the M.S.E. and Ph.D. degrees in electrical engineering from Stanford University, Stanford, CA, in 1994 and 1999, respectively, with a minor in business administration. He received the B.S.E. degree (highest honor) in electrical engineering from Princeton University, Princeton, NJ, in 1993, with certificates in applied and computational mathematics, engineering physics, and engineering and management systems.

He is currently an Associate Professor with the Department of Computer Science and Engineering, The Hong Kong University of Science and Technology (HKUST), Kowloon, Hong Kong. His research interest includes multimedia networking, peer-to-peer technologies and streaming, and wireless communication networks. He is the Director of Sino Software Research Institute (SSRI) at the HKUST, was a Visiting Research Collaborator at Princeton University (2009), a Visiting Associate Professor at Stanford University (2008–09), the Director of the Computer Engineering Program at the HKUST (2006–2008), a Visiting Assistant Professor in Networking at the University of California at Davis (1998–1999), and Research Intern at the NEC Research Institute, Princeton, NJ (1992–1993).

Dr. Chan is an Associate Editor of the IEEE TRANSACTIONS ON MULTIMEDIA and a Vice-Chair of the Peer-to-Peer Networking and Communications Technical Sub-Committee and the IEEE Comsoc Emerging Technologies Committee. He is also a Guest Editor of the Special Issue on Interactive Multimedia, IEEE TRANSACTIONS ON MULTIMEDIA (2011), and Special Issue on Distributed Image Processing and Communications, *IEEE Signal Processing Magazine* (2011). He was the TPC chair of the IEEE Consumer Communications and Networking Conference (CCNC) in 2010. From 2003–2010, he served as a Vice-Chair of the IEEE COMSOC Multimedia Communications Technical Committee (MMTC). He was a Guest Editor of the Special Issue on Peer-to-Peer Multimedia Streaming in the *IEEE Communication Magazine* (2007) and Advances in Consumer Communications and Networking in Springer Multimedia Tools and Applications (2007). He was a co-chair of the multimedia symposium in IEEE Globecom (2007 and 2006) and IEEE ICC (2007 and 2005), and of the workshop on Advances in Peer-to-Peer Multimedia Streaming in ACM Multimedia Conference (2005). He is a member of honor societies Tau Beta Pi, Sigma Xi, and Phi Beta Kappa. He was a William and Leila Fellow at Stanford University (1993–1994). At Princeton, he was the 1993 recipient of the Charles Ira Young Memorial Tablet and Medal and the POEM Newport Award of Excellence.

Xing Jin received the B.Eng. degree in computer science and technology from Tsinghua University, Beijing, China, in 2002, and the Ph.D. degree in computer science and engineering from The Hong Kong University of Science and Technology (HKUST), Kowloon, Hong Kong, in 2007.

He is currently a Software Engineer with the Data Infrastructure Team, Facebook, Inc., Palo Alto, CA. His research interests include distributed information storage and retrieval, peer-to-peer technologies, multimedia networking, and Internet topology inference.

Dr. Jin is a member of Sigma Xi and the IEEE COMSOC Multimedia Communications Technical Committee. He has been on the editorial board of the *Journal of Multimedia* since 2006, and the *Canadian Journal of Pure and Applied Sciences* since 2007. He was awarded the Microsoft Research Fellowship in 2005.