Optimal Multipath Routing for Performance Guarantees in Multi-hop Wireless Networks

Sastry Kompella[*]   Shiwen Mao[§]   Y. Thomas Hou[†]   Hanif D. Sherali[‡]

[*]Information Technology Division, Naval Research Laboratory, Washington, DC 20375

[§]Department of Electrical and Computer Engineering, Auburn University, Auburn, AL 36849

[†]The Bradley Department of Electrical and Computer Engineering, Virginia Tech, Blacksburg, VA 24061

[‡]The Grado Department of Industrial and Systems Engineering, Virginia Tech, Blacksburg, VA 24061

Abstract—**In this paper, we consider the problem of optimal multipath routing for providing application performance guarantees in multi-hop wireless networks, using multiple description video streaming as our target application. We address this problem, which is shown to be NP-hard, with a novel *Reformulation-Linearization Technique* (RLT) and *branch-and-bound*-based approach, and develop an algorithm that produces a pair of paths within the $(1 - \epsilon)$ range of the global optimum. The proposed algorithm is computationally efficient and this $(1 - \epsilon)$-optimal algorithm provides an elegant tradeoff between optimality and computational complexity.**

I. INTRODUCTION

Recently, there has been considerable interest in supporting video applications in multi-hop wireless networks (e.g., ad hoc networks or mesh networks). The dynamic characteristics associated with multi-hop wireless networks (e.g., mobility, topology change, radio fading and loss) have posed some unique challenges for video communications. Existing routing protocols for such networks (e.g., OLSR [5], DSR [8], AODV [12]) mainly focus on network layer connectivity problem and do not have explicit consideration for video application requirements. As a result, such single layer (network layer) approaches are not optimal to support video applications.

In this paper, we investigate the important problem of how to design routing protocols so as to optimally support video applications in multi-hop wireless networks. We aim to develop a cross-layer approach where the routing decision takes explicit consideration of the application layer objective function. In other words, the routing engine at the network layer will not only find routes for connectivity, but will find optimal routes such that the application layer performance metric (e.g., PSNR) is maximized.

Throughout this investigation, we will employ *multiple description* (MD) video as our target application. MD video is an important coding technique for error resilience and control for multimedia applications [15] and has been recognized as an ideal candidate for video streaming in multi-hop wireless networks [10]. Under MD coding, multiple *equivalent* streams (or descriptions) are generated for a video source for transmission. At the receiver, *any* received subset of these descriptions can be combined to reconstruct the original video and the quality of the reconstructed video is commensurate with the number of received descriptions. This video coding technique is drastically different from traditional layered video coding,

where video reconstruction hinges upon successful delivery of the base layer.

From cross-layer routing perspective, the problem is to find a set of routes (or paths) in multi-hop wireless networks, one for each video description such that the video distortion is minimized. The optimal multipath routing problem considered in this paper is formulated into a *mixed-integer non-linear programming (NLP) problem*. Such problems are shown to be NP-hard in general [14]. In a previous work [9], we studied this problem and addressed it using *Genetic Algorithms* (GA). Although GA is an effective algorithm, it nevertheless is a metaheuristic which does not provide any performance bounds on how close the solution is to the optimal. As a result, a theoretical result for multipath routing for MD video remains an open problem.

In this paper, we aim to bridge this important theoretical gap in cross-layer optimization for video communications. We present a formal solution procedure based on the so-called *branch-and-bound* [11] framework, which aims to produce an $(1 - \epsilon)$-optimal solution. Here $\epsilon > 0$ is an arbitrarily small number reflecting required accuracy. A key component in the solution procedure is a novel method called the *Reformulation-Linearization Technique* (RLT) [13], which in essence produces tight lower bound for a minimization problem. A global optimal solution can be obtained by embedding RLT into the branch-and-bound framework. That is, during each branch-and-bound iteration, we employ RLT to generate an LP relaxation for the corresponding sub-problem, and by solving this LP relaxation we generate a lower bound for the sub-problem. The corresponding upper bound is computed by a suitable local search algorithm on the solution provided by the LP relaxation. Based on the bounds of all the sub-problems currently in the branch-and-bound tree, the gap between the lower and upper bounds of the original problem is narrowed iteratively, until they are within ϵ of each other.

The remainder of this paper is organized as follows. In Section II, we present the problem formulation. In Section III, we describe an RLT-based approach to reformulate and linearize the problem. Section IV presents a branch-and-bound-based solution procedure. Simulation results are presented in Section V, and Section VI concludes this paper.

II. PROBLEM FORMULATION

A. Network Model

We model a multi-hop wireless network as a directed graph $\mathcal{G}\{\mathcal{V}, \mathcal{E}\}$, where $\mathcal{V}$ is the set of vertexes representing wireless nodes and $\mathcal{E}$ the set of edges representing wireless links.

We assume that nodes are reliable during the video session, but links may be up or down with certain probabilities. For our routing problem, we focus on network layer statistics, assuming that the physical and MAC layer dynamics from the underlying radio environment are reflected in these metrics. We characterize a link $\{i, j\} \in \mathcal{E}$ with (i) b_{ij}: the available bandwidth of link $\{i, j\}$; (ii) p_{ij}: the probability that link $\{i, j\}$ is "up"; (iii) l_{ij}: average burst length for packet losses on link $\{i, j\}$. Based on these basic metrics, we can derive path-level bandwidth and failure probability, which are useful to characterize end-to-end performance at the video application layer (i.e., distortion).

B. Video Distortion and Path Level Statistics

1) Video Distortion: Consider a video session from video server s to client t. We assume that the video is encoded into two descriptions, i.e., double description (DD) video, each with a rate R_h bits/pixel, $h = 1, 2$. We consider DD video since it is most widely used for MD video [2]–[4], [10]. Let d_h be the achieved distortion when only description h is received, $h = 1, 2$, and d_0 the distortion when both descriptions are received. In this paper, we employ the following distortion-rate function [1], [9]:

$$\begin{cases} d_0 = \frac{2^{-2(R_1+R_2)}}{2^{-2R_1} + 2^{-2R_2} - 2^{-2(R_1+R_2)}} \cdot \sigma^2 \\ d_1 = 2^{-2R_1} \cdot \sigma^2 \\ d_2 = 2^{-2R_2} \cdot \sigma^2, \end{cases} \quad (1)$$

where σ^2 is the variance of the source.

From end-to-end perspective, let π_{00} denote the probability of receiving both descriptions, π_{01} the probability of receiving description 1 only, π_{10} the probability of receiving description 2 only, and π_{11} the probability of losing both descriptions. Then, the expected average video distortion at the receiver can be approximated as: $D = \pi_{00} \cdot d_0 + \pi_{01} \cdot d_1 + \pi_{10} \cdot d_2 + \pi_{11} \cdot \sigma^2$. Note that our solution procedure presented later in this paper does not depend on the specific structure of the distortion-rate function.

2) Path-Level Statistics: To characterize a path $\mathcal{P}_h$ between source node s and destination node t, we define:

$$I_{ij}^{(h)} = \begin{cases} 1, & \text{if link } \{i, j\} \in \mathcal{P}_h, \\ 0, & \text{otherwise.} \end{cases}$$
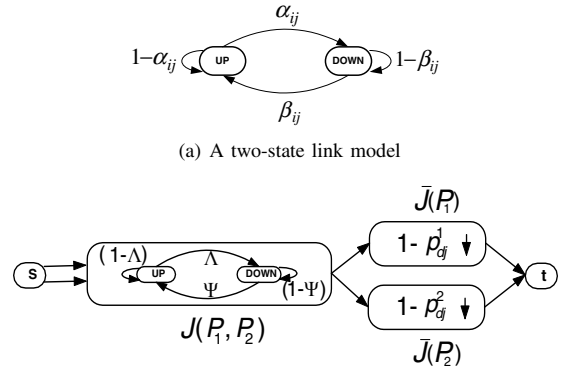
An arbitrary path $\mathcal{P}_h$ can then be represented by a vector $\mathbf{I}^{(h)}$ of $|\mathcal{E}|$ elements, each corresponding to a link and having a binary value.

For a source-destination pair $\{s, t\}$, consider two given paths $[\mathcal{P}_1, \mathcal{P}_2]$ in $\mathcal{G}\{\mathcal{V}, \mathcal{E}\}$. Since we do not mandate "disjointedness" between the two paths, $\mathcal{P}_1$ and $\mathcal{P}_2$ may share nodes and links. For each link $\{i, j\}$, the aggregate description rate should be bounded by its available bandwidth as

$$I_{ij}^{(1)} \cdot R_1 + I_{ij}^{(2)} \cdot R_2 \leq \rho \cdot b_{ij},$$

where ρ is a constant. For a video with coding rate f frames/s and a resolution of $W \times V$ pixels/frame, we have $\rho = 1/(\kappa \cdot W \cdot V \cdot f)$, where κ is a constant determined by the chroma sub-sampling scheme (e.g, $\kappa = 1.5$ for QCIF).

We now focus on how to compute the end-to-end path statistics. Similar to the approach in [2], [3], we classify the links into three sets: set one consisting of links shared by both paths, denoted as $\mathcal{J}(\mathcal{P}_1, \mathcal{P}_2)$, and the other two sets consisting

(a) A two-state link model

(b) A path model characterizing joint and disjoint links

Fig. 1. Link and path models.

of disjoint links on the two paths, denoted as $\bar{\mathcal{J}}(\mathcal{P}_h)$, $h = 1, 2$, respectively. For disjoint portion of the paths, it suffices to model the packet loss as a Bernoulli event, since losses of the two descriptions are assumed to be independent on disjoint portions. The success probabilities on the disjoint portions are:

$$p_{dj}^{(h)} = \begin{cases} \prod_{\{i,j\} \in \bar{\mathcal{J}}(\mathcal{P}_h)} p_{ij}, & \text{if } \bar{\mathcal{J}}(\mathcal{P}_h) \neq \emptyset, h = 1, 2 \\ 1, & \text{otherwise, } h = 1, 2. \end{cases} \quad (2)$$

On the joint portion of the paths, losses on the two streams are correlated. In order to model such correlation, we model each shared link $\{i, j\}$ as an on-off process modulated by a discrete-time Markov chain, as shown in Figure 1(a). There is no packet loss when the link is "up"; all packets are dropped when the link is "down". Transition probabilities, $\{\alpha_{ij}, \beta_{ij}\}$, can be computed from the link statistics as $\beta_{ij} = 1/l_{ij}$ and $\alpha_{ij} = (1 - p_{ij})/(p_{ij}l_{ij})$.

If there are K shared links, the aggregate failure process of these links is a Markov process with 2^K states. In order to simplify the computation, we model the aggregate process as an on-off process. Specifically, we lump up all the states with at least one link failure into a single "down" state, while using the remaining state where all the links are in good condition as the "up" state. Let T_{on} be the average length of the "up" period. We have $T_{on} = 1/[1 - \prod_{\{i,j\} \in \mathcal{J}(\mathcal{P}_1, \mathcal{P}_2)}(1 - \alpha_{ij})]$ The transition probabilities of the aggregate on-off process can be computed as $\Lambda = 1/T_{on}$, and $\Psi = p_{jnt}/[T_{on}(1 - p_{jnt})]$, where p_{jnt} is the average success probability of the joint portion, and

$$p_{jnt} = \begin{cases} \prod_{\{i,j\} \in \mathcal{J}(\mathcal{P}_1, \mathcal{P}_2)} p_{ij}, & \text{if } \mathcal{J}(\mathcal{P}_1, \mathcal{P}_2) \neq \emptyset \\ 1, & \text{otherwise.} \end{cases} \quad (3)$$

Note that $\Lambda = 0$ and $\Psi = 0$ if $\mathcal{J}(\mathcal{P}_1, \mathcal{P}_2) = \emptyset$.

The consolidated path model is illustrated in Figure 1(b), where $\mathcal{J}(\mathcal{P}_1, \mathcal{P}_2)$ is modeled as a two-state Markov process with parameters $\{\Lambda, \Psi\}$, and $\bar{\mathcal{J}}(\mathcal{P}_h)$ is modeled as a Bernoulli process with parameter $(1 - p_{dj}^{(h)})$, $h = 1, 2$. With the consolidated path model, the joint probabilities of receiving the descriptions are as follows.

$$\begin{cases} \pi_{00} = p_{jnt} \cdot (1 - \Lambda) \cdot p_{dj}^{(1)} \cdot p_{dj}^{(2)} \\ \pi_{01} = p_{jnt} \cdot p_{dj}^{(1)} \cdot \left[1 - (1 - \Lambda) \cdot p_{dj}^{(2)} \right] \\ \pi_{10} = p_{jnt} \cdot \left[1 - (1 - \Lambda) p_{dj}^{(1)} \right] \cdot p_{dj}^{(2)} \\ \pi_{11} = 1 - p_{jnt} \cdot \left[p_{dj}^{(1)} + p_{dj}^{(2)} - (1 - \Lambda) \cdot p_{dj}^{(1)} \cdot p_{dj}^{(2)} \right] \end{cases} \quad (4)$$

C. Problem Formulation

We can now formulate the problem of multipath routing for MD video into the following mathematical programming.

In Problem OPT-MR, $\{I_{ij}^{(h)}\}$ are binary optimization variables (incorporated in π_{00}, π_{01}, π_{10}, and π_{11}). Constraint (6) guarantees that the paths originate at the source s and terminate at the destination t, and constraint (7) ensures that the paths are loop-free. Constraint (8) guarantees that the links are stable. For a given pair of paths, the average video distortion D is determined by the end-to-end statistics and the correlation of the paths, as given in (1) and (4).

OPT-MR

Minimize:
$$D = \pi_{00} \cdot d_0 + \pi_{01} \cdot d_1 + \pi_{10} \cdot d_2 + \pi_{11} \cdot \sigma^2 \quad (5)$$

subject to:
$$\sum_{j \in \mathcal{V}} I_{ij}^{(h)} - \sum_{j \in \mathcal{V}} I_{ji}^{(h)}$$
$$= \begin{cases} 1, & \text{if } i = s, \quad i \in \mathcal{V}, \ h = 1, 2 \\ -1, & \text{if } i = t, \quad i \in \mathcal{V}, \ h = 1, 2 \\ 0, & \text{otherwise}, \end{cases} \quad (6)$$

$$\sum_{i \in \mathcal{V}} I_{ij}^{(h)} \begin{cases} \leq 1, & \text{if } i \neq t, \quad j \in \mathcal{V}, \ h = 1, 2 \\ = 0, & \text{if } i = t, \quad j \in \mathcal{V}, \ h = 1, 2 \end{cases} \quad (7)$$

$$I_{ij}^{(1)} \cdot R_1 + I_{ij}^{(2)} \cdot R_2 \leq \rho \cdot b_{ij}, \quad \{i, j\} \in \mathcal{E} \quad (8)$$

$$I_{ij}^{(h)} \in \{0, 1\}, \quad \{i, j\} \in \mathcal{E}, \ h = 1, 2. \quad (9)$$

The objective function (5) is a complex ratio of high-order exponentials of the I-variables. The objective evaluation of a pair of paths involves identifying the joint and disjoint portions, which is only possible when both paths are completely determined. Since such problems are NP-hard in general [7], and Problem OPT-MR does not appear to posses any special simplifying structure, it is likely to be NP-hard, although a formal proof is not given in this paper.

III. REFORMULATION AND LINEARIZATION

Our solution approach to problem OPT-MR is to embed a the novel reformulation-linearization technique (RLT) in a branch-and-bound framework [13]. RLT is a relaxation technique that can be used to produce tight polyhedral outer approximations or linear programming relaxations for an underlying non-linear, non-convex polynomial programming problem. In the following, we first reformulate problem OPT-MR into a *mixed-integer polynomial programming problem* P-MR. Then, we replace all the non-linear terms and add the corresponding RLT constraints into the problem formulation, so as to obtain a *linear programming relaxation* of problem OPT-MR, denoted as L-MR. In the next section, we will develop a branch-and-bound based procedure that finds $(1-\epsilon)$-optimal solution.

A. Reformulating Problem OPT-MR

As discussed, the objective function of problem OPT-MR is a complex function of exponential terms of the I-variables. Our first goal is to reformulate these terms, which will simplify the objective function and the constraints. Without loss of generality, we set $\sigma^2 = 1$ to simplify notation. Note that σ^2 only affects the absolute value of distortion, but not optimal routing selection.

In (4), there are four high order terms that need to be reformulated, namely, p_{jnt}, $p_{dj}^{(1)}$, $p_{dj}^{(2)}$, and Λ. In the following, we show how to reformulate a few of these terms. From their definitions in (2) and (3), we can rewrite the success probabilities as:

$$\begin{cases} p_{jnt} = \prod_{\{i,j\} \in \mathcal{E}} p_{ij}^{\{I_{ij}^{(1)} \cdot I_{ij}^{(2)}\}} \\ p_{dj}^{(1)} = \prod_{\{i,j\} \in \mathcal{E}} p_{ij}^{\{I_{ij}^{(1)} \cdot (1 - I_{ij}^{(2)})\}} \\ p_{dj}^{(2)} = \prod_{\{i,j\} \in \mathcal{E}} p_{ij}^{\{I_{ij}^{(2)} \cdot (1 - I_{ij}^{(1)})\}}. \end{cases} \quad (10)$$

Taking logarithms on both sides, we can convert the high order terms on the right-hand-side (RHS) of (10) into summations of quadratic terms of the I-variables, i.e.,

$$\begin{cases} \log(p_{jnt}) = \sum_{\{i,j\} \in \mathcal{E}} \left[I_{ij}^{(1)} \cdot I_{ij}^{(2)} \cdot \log(p_{ij}) \right] \\ \log(p_{dj}^{(1)}) = \sum_{\{i,j\} \in \mathcal{E}} \left[I_{ij}^{(1)} \cdot (1 - I_{ij}^{(2)}) \cdot \log(p_{ij}) \right] \\ \log(p_{dj}^{(2)}) = \sum_{\{i,j\} \in \mathcal{E}} \left[I_{ij}^{(2)} \cdot (1 - I_{ij}^{(1)}) \cdot \log(p_{ij}) \right]. \end{cases} \quad (11)$$

Other non-linear terms can be reformulated in a similar way. Having simplified the high-order terms, we now deal with the resulting constraints of the form $y = \log(\lambda)$. We can linearize this logarithmic relationship over some tightly-bounded interval using a *polyhedral outer approximation* comprised of a *convex envelope* in concert with several tangential supports. For instance, if λ is bounded as $0 < \lambda_0 \leq \lambda \leq 1$, these constraints can be written as follows.

$$\begin{cases} y \geq \frac{\log(\lambda_0)}{1 - \lambda_0} \cdot (1 - \lambda) \\ y \leq \log(\lambda_k) + \frac{\lambda - \lambda_k}{\lambda_k}, \quad k = 1, ..., k_{max}, \end{cases} \quad (12)$$

where $\lambda_k = \lambda_0 + (1 - \lambda_0) \cdot (k - 1)/(k_{max} - 1)$, for $k = 1, 2, \cdots, k_{max}$. A four-point tangential approximation can be obtained by letting $k_{max} = 4$. The corresponding convex envelope consists of a chord connecting the two end points, which is used in combination with tangential supports at four points including the two end points. As a result, every logarithmic relationship translates to five linear constraints as shown in (12). Note that such polyhedral outer approximations will be iteratively tightened during the branch-and-bound procedure (see Section IV). This reduces OPT-MR into a *mixed-integer polynomial programming problem* P-MR.

B. Linearizing Problem P-MR

Although greatly simplified, problem P-MR is still a polynomial programming problem, which is NP-hard in general [13]. In this section, we linearize problem P-MR by employing RLT, which involves variable substitutions and introducing linear RLT bound-factor constraints.

Consider a quadratic product term of the form $(p_{dj}^{(1)} \cdot p_{dj}^{(2)})$ in (4). By introducing a new variable $z_0 = p_{dj}^{(1)} \cdot p_{dj}^{(2)}$, we can substitute the $(p_{dj}^{(1)} \cdot p_{dj}^{(2)})$ terms with z_0, thus removing this quadratic term from the objective function and constraints. Assuming $p_{dj}^{(1)}$ and $p_{dj}^{(2)}$ are each bounded as $\left(p_{dj}^{(1)}\right)_L \leq p_{dj}^{(1)} \leq \left(p_{dj}^{(1)}\right)_U$ and $\left(p_{dj}^{(2)}\right)_L \leq p_{dj}^{(2)} \leq \left(p_{dj}^{(2)}\right)_U$, respectively, we can add the following relational constraints, which are known as

the *RLT bound-factor product constraints*:

$$
\left\{
\begin{array}{l}
\left\{ \left[p_{dj}^{(1)} - \left(p_{dj}^{(1)} \right)_L \right] \cdot \left[p_{dj}^{(2)} - \left(p_{dj}^{(2)} \right)_L \right] \right\}_{LS} \geq 0 \\[2mm]
\left\{ \left[p_{dj}^{(1)} - \left(p_{dj}^{(1)} \right)_L \right] \cdot \left[\left(p_{dj}^{(2)} \right)_U - p_{dj}^{(2)} \right] \right\}_{LS} \geq 0 \\[2mm]
\left\{ \left[\left(p_{dj}^{(1)} \right)_U - p_{dj}^{(1)} \right] \cdot \left[p_{dj}^{(2)} - \left(p_{dj}^{(2)} \right)_L \right] \right\}_{LS} \geq 0 \\[2mm]
\left\{ \left[\left(p_{dj}^{(1)} \right)_U - p_{dj}^{(1)} \right] \cdot \left[\left(p_{dj}^{(2)} \right)_U - p_{dj}^{(2)} \right] \right\}_{LS} \geq 0,
\end{array}
\right.
$$

where $\{\cdot\}_{LS}$ denotes a *linearization step* under the substitution $z_0 = p_{dj}^{(1)} \cdot p_{dj}^{(2)}$. Expanding the above inequalities and substituting $z_0 = p_{dj}^{(1)} \cdot p_{dj}^{(2)}$, we obtain RLT constraints for z_0. By adding the linear RLT bound-factor constraints for z_0 into the problem formulation, we can therefore replace the second-order term $p_{dj}^{(1)} \cdot p_{dj}^{(2)}$ with the linear term z_0.

Similarly, we define new variables for all the remaining non-linear terms in the reformulated problem OPT-MR(p), including $z_1 = p_{jnt} \cdot p_{dj}^{(1)}$, $z_2 = p_{jnt} \cdot p_{dj}^{(2)}$, and $z_3 = z_0 \cdot \phi$, and make substitutions in the same manner. The constraints derived from reformulating the logarithmic terms can also be linearized by substituting $z_{ij} = I_{ij}^{(1)} \cdot I_{ij}^{(2)}$, and by introducing the corresponding linear RLT bound-factor constraints, for all $\{i, j\} \in \mathcal{E}$. As a result, we obtain a *linear programming relaxation* problem L-MR, which can be solved in polynomial-time.

IV. A SOLUTION PROCEDURE

A. Overview of the Branch-and-Bound Framework

Branch-and-bound is an algorithmic method for solving optimization problems, especially in discrete and combinatorial optimization [13]. Under branch-and-bound, the original problem is first relaxed using a suitable *relaxation technique* to obtain an easier-to-solve, lower-bounding problem. In our approach, we used RLT to reformulate and linearize OPT-MR into an LP relaxation L-MR. The optimal solution to this LP relaxation provides a lower bound LB for the original problem. Since such an LP relaxation usually yields an infeasible solution to the original problem, a *local search algorithm* should be employed to obtain a feasible solution to the original problem. The resulting feasible solution then provides an upper bound UB for the original problem.

Under branch-and-bound framework, the original problem O (or OPT-MR) is partitioned into sub-problems, each having a smaller feasible solution space, based on the solution provided by the LP relaxation. New sub-problems are organized as a *branch-and-bound tree*, while this partitioning or branching process is carried out recursively to obtain two new sub-problems at each node of the tree. The sub-problems are also inserted into a *problem list* L, which records the active nodes in the branch-and-bound tree structure. More specifically, in the beginning, the problem list L is initialized with the original problem O. At any given iteration, the lower and upper bounds for O are computed as

$$
\left\{
\begin{array}{l}
LB = \min\{LB_k : \text{Problem } k \in L\} \\
UB = \min\{UB_k : \text{all nodes } k \text{ explored thus far}\}.
\end{array}
\right.
\tag{13}
$$

The method proceeds by choosing the next problem to partition from the problem list. In our approach, the problem $k \in L$ having the smallest LB_k is chosen. This problem k is then partitioned into two sub-problems k_1 and k_2, which

replace problem k in L. Every time a problem k is added to the list, LB_k and UB_k are computed, and the LB and UB for the original problem O are updated. At any given iteration, if $LB \geq (1 - \epsilon) \cdot UB$, the procedure terminates and we have an $(1 - \epsilon)$-optimal solution. Also, for any problem k in the problem list, if $LB_k \geq (1 - \epsilon) \cdot UB$, no globally optimal solution that improves beyond the ϵ-tolerance can exist in the sub-space of the feasible region represented by this node. Therefore, this node can be removed (or *fathomed*) from the branch-and-bound tree. In this manner, the branch-and-bound process can fathom certain branches or nodes of the tree, eliminating them from further exploration. The effectiveness of the branch-and-bound procedure depends strongly on that of the employed fathoming strategy.

B. Details of the Solution Procedure

We now describe the details of the solution procedure, which we call ALG(ϵ). The iterative branch-and-bound algorithm terminates when either the lower bound for the original problem is within $(1 - \epsilon)$ of the upper bound, i.e., $LB \geq (1 - \epsilon) \cdot UB$, or the problem list L is empty. The operation of each step in ALG(ϵ) is described in the following.

1) Initialization and Relaxation : We start by initializing the current "best" solution, denoted as ψ^*, with any known feasible solution $\bar{\psi}$ (such as a pair of shortest paths or disjoint shortest paths) and the current "best" upper bound UB as the objective value obtained using this solution $\bar{\psi}$.

2) Node Selection : At every iteration, problem k (or the corresponding node in the branch-and-bound tree) that has the minimum LB_k among all the problems $k \in L$ is selected. As discussed before, this problem is indicative of the lower bound for the original problem. Subsequent operations of local search, partitioning and bounding are performed on this problem k.

3) Local Search : As discussed in Section IV-A, the solution to the relaxation problem k that is selected in the node selection step, is usually infeasible to the original problem O. This is especially true if the original problem involves binary variables (i.e., the I-variables could be fractions). A local search algorithm should be used to find a feasible solution to the original problem starting from the infeasible lower bounding solution.

Let $\hat{\psi}$ be the infeasible (or fractional) solution obtained by solving the LP relaxation of the original problem. Starting from this fractional solution, we solve for $h = 1, 2$ the following shortest path problem: **Minimize** $\sum_{\{i,j\} \in \mathcal{E}} \left[-\hat{I}_{ij}^{(h)} \right] \cdot I_{ij}^{(h)}$, subject to the flow constraints. Note that for an optimization variable y, $\hat{y}$ denotes its value in the infeasible solution $\hat{\psi}$. Solving these shortest path problems provides us with a rounded heuristic solution $\bar{\psi}$ that has a tendency to round up relatively higher-valued components of $\hat{\psi}$ and round down relatively lower-valued components. The distortion value of the rounded solution $\bar{\psi}$ is an upper bound for this subproblem, i.e., UB_k.

4) Partitioning : The objective of the partitioning step is to find the branching variable that will enable us to split the feasible solution space Ω_k of problem k into two solution sub-spaces Ω_{k_1} and Ω_{k_2}. In ALG(ϵ), we need to consider three classes of optimization variables for partitioning, i.e., the

binary I-variables, the substitution variables (e.g., z_0), and the logarithm substitution terms.

When partitioning based on the I-variables, we need to select a variable that will offer the highest gain in terms of improving the objective value. For this purpose, we should choose the I-variable which is factional and the closest to 0.5, and partition the problem by fixing it to 0 and 1.

In addition to the I-variables, we also need to examine branching decisions based on the substitution variables such as $z_0 = p_{dj}^{(1)} \cdot p_{dj}^{(2)}$. For such variables, we first find the maximum relaxation error between the substitution variable and the corresponding product term, say, $|\hat{p}_{dj}^{(1)} \cdot \hat{p}_{dj}^{(2)} - \hat{z}_0|$. We then verify whether the following condition is satisfied:

$$\left[\left(p_{dj}^{(1)}\right)_U - \left(p_{dj}^{(1)}\right)_L\right] \min\left[\hat{p}_{dj}^{(1)} - \left(p_{dj}^{(1)}\right)_L, \left(p_{dj}^{(1)}\right)_U - \hat{p}_{dj}^{(1)}\right] \geq$$
$$\left[\left(p_{dj}^{(2)}\right)_U - \left(p_{dj}^{(2)}\right)_L\right] \min\left[\hat{p}_{dj}^{(2)} - \left(p_{dj}^{(2)}\right)_L, \left(p_{dj}^{(2)}\right)_U - \hat{p}_{dj}^{(2)}\right].$$

If this condition holds true, we partition the solution space Ω_k of problem k into two new regions Ω_{k_1} and Ω_{k_2}, by dividing the range $\left[\left(p_{dj}^{(1)}\right)_L, \left(p_{dj}^{(1)}\right)_U\right]$ into two subregions $\left[\left(p_{dj}^{(1)}\right)_L, \hat{p}_{dj}^{(1)}\right]$ and $\left(\hat{p}_{dj}^{(1)}, \left(p_{dj}^{(1)}\right)_U\right]$. Otherwise, we partition Ω_k with respect to $p_{dj}^{(2)}$ in a similar manner.

Finally, the branching decisions also include the logarithm substitution terms, as described earlier. In such cases, we first find the variable that gives the greatest discrepancy between the logarithm value, say, $\log(\hat{\phi})$ and the RHS of its corresponding substitution among all such terms, and then either bisect the interval of this variable (e.g., $[(\phi)_L, (\phi)_U]$) evenly, or divide this interval at the point $\hat{\phi}$.

5) Bounding : In the bounding step, we solve the RLT relaxation for the two sub-problems identified in the partitioning step, and obtain their corresponding lower bounds LB_{k_1} and LB_{k_2}, thereby updating the incumbent lower bounding solution. The corresponding upper bounds, i.e., UB_{k_1} and UB_{k_2}, are obtained by applying the local search algorithm starting from the relaxation solutions obtained, and the current LB and UB values for the original problem O are updated according to (13). If any of the conditions $(1-\epsilon) \cdot UB > LB_{k_1}$ and $(1-\epsilon) \cdot UB > LB_{k_2}$ are satisfied, we add the corresponding problem into the problem list L, and remove problem k from the list.

6) Fathoming : For any problem k in the problem list L, if $LB_k \geq (1-\epsilon) \cdot UB$, then the sub-space corresponding to this problem does not contain any solution that improves beyond the ϵ-tolerance of the incumbent solution. Therefore, we can prune this problem from the problem list.

V. SIMULATION STUDIES

Our simulation study consists of two parts. In the first part, we will examine the convergence behavior and complexity issue of the proposed algorithm. In the second part, we will demonstrate the performance advantage of the proposed cross-layer approach over a non-cross-layer approach. Throughout our simulation study, we consider a multi-hop wireless network deployed over a rectangular region, where the connectivity between the nodes is determined by the radio transmission range. The source node s and destination node t are chosen randomly from the nodes in the network. At the link level, we associate each link with a failure probability,

TABLE I
PERFORMANCE OF THE PROPOSED ALGORITHM ($\epsilon = 0.01$)

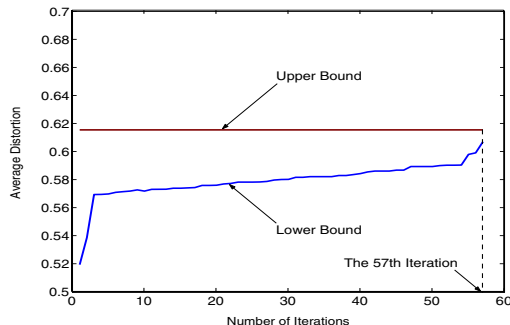| Number of Nodes	Mean Computation Time(sec)	Variance
20	0.074	0.002
30	0.381	0.282
50	0.858	0.460

Fig. 2. Convergence behavior for a 50-node network.

available bandwidth, and mean burst length (for packet loss). Specifically, the failure probability is taken uniformly between $[0.01, 0.3]$; the available bandwidth is taken uniformly from the set $[100, 150, 200, 250, 300, 350, 400]$ Kb/s; and the mean burst length is chosen uniformly between $[2,6]$. For ALG(ϵ), we set $\epsilon = 0.01$ (or 1%). We implement the BB/RLT solution procedure in C program and use the LINDO API 3.0 for solving the LP relaxation problem.

A. Convergence Behavior of Solution Procedure

We first examine the convergence behavior of the solution procedure for different network sizes and topologies. We consider 20-, 30- and 50-node networks within a 300m x 300m, 400m x 400m, and 500m x 500m rectangular regions, respectively. The transmission range for each node is assumed to be 150m. The description rates are $R_1 = R_2 = 128$ Kb/s. For each network size, we generate 100 topologies and run 100 computations to obtain the mean and variance of convergence time. Table I shows the convergence time performance for the networks with $\epsilon = 0.01$. The ALG(ϵ) algorithm was run on a standard desktop PC with a Pentium-4 2.4 GHz processor and 512 MB memory. As shown in Table I, the computational time for convergence to $\epsilon = 0.01$ is very fast for small to moderate sized network.

To see the iterative convergence behavior of the solution procedure, we intentionally pick an experiment with a convergence time longer than the average. Figure 2 shows such an instance for the 50-node network for description rates of $R_1 = R_2 = 320$ Kb/s. For this particular 50-node network, the gap between upper bound UB and lower bound LB converges to $\epsilon = 1\%$ (i.e., $LB \geq 99\% \cdot UB$) after the 57th iteration in about 1.87 s.

B. Comparison with k-Shortest Path Routing

In this section, we compare our cross-layer routing approach with a popular non-cross-layer approach. For the latter, we consider the k-shortest path (SP) routing algorithm [6], with $k = 2$ or 2-SP for DD video. We use hop count as the routing metric in the 2-SP algorithm.

| (a) Original | (b) 128 Kb/s, ALG(ϵ) | (c) 128 Kb/s, 2-SP | (d) 256 Kb/s, ALG(ϵ) | (e) 256 Kb/s, 2-SP |

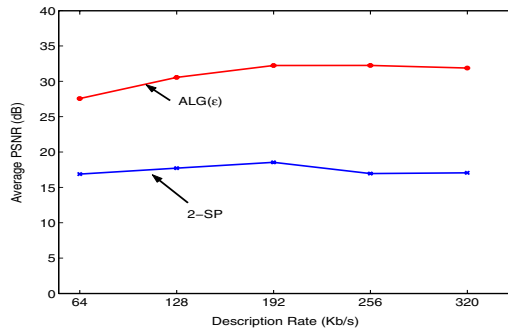Fig. 3.   Frame 278 from the reconstructed video.

Fig. 4.   Average PSNR values for different description rates.

We encode a video sequence in order to transmit the DD video over the network, and compare the video quality at the receiver (measured using PSNR) under our cross-layer approach and the 2-SP approach. An H.263+ like codec is implemented to generate the two descriptions. This codec encodes the video sequence into two balanced descriptions (i.e., $R_1 = R_2$). The QCIF sequence "Foreman" (400 frames) is encoded at 15 fps for each description, with a macroblock level intra-refreshment rate of 10%. Each Group of Blocks (GOB) is carried in a different packet. When a GOB is corrupted, the decoder applies a simple error concealment scheme by copying the corresponding slice from the most recent, correctly received frame. We vary the rate of each video description from 64 Kb/s to 320 Kb/s for the 50-node network and compare the PSNR performance under our solution and 2-SP. The results are shown in Figure 4. Note that as the description rate R increases, more links will become ineligible during path selection process. We find that our cross-layer approach provides higher PSNR over the 2-SP approach under all description rates.

To illustrate the quality of video frames, we plot a sample video frame from the the original video in Figure 3(a) and compare it to the reconstructed video frames under our ALG(ϵ) and 2-SP for $R = 128$ Kb/s and $R = 256$ Kb/s, respectively in Figure 3(b)-(e). The frames under ALG(ϵ) have a visual quality very close to the original frame, while the frames under 2-SP are barely recognizable.

VI. CONCLUSIONS

In this paper, we studied the problem of how to route MD video over multi-hop wireless networks with the objective of optimizing the application layer performance. We formulated this problem into an cross-layer optimization problem with an application performance metric as the objective function and routing and link layer considerations as constraints. We developed a formal branch-and-bound solution procedure and employed a novel relaxation technique called RLT in the solution procedure. We showed that this solution procedure is able to produce a set of routes with an objective value within $(1 - \epsilon)$ of the optimal value. Simulation results demonstrated the efficacy of the proposed solution procedure.

REFERENCES

[1] M. Alasti, K. Sayrafian-Pour, A. Ephremides, and N. Farvardin. Multiple description coding in networks with congestion problem *IEEE Trans. Information Theory*, 47(3):891–902, March 2001.
[2] J.G. Apostolopoulos, T. Wong, W. Tan, and S. Wee. On multiple description streaming in content delivery networks. In *Proc. IEEE INFOCOM*, pp. 1736–1745, New York, NY, June 2002.
[3] A.C. Begen, Y. Altunbasak, O. Ergun, and M.H. Ammar. Multi-path selection for multiple description encoded video streaming. *EURASIP Signal Processing: Image Commun.*, 20(1):39–60, Jan. 2005.
[4] J. Chakareski, S. Han, and B. Girod. Layered coding vs. multiple descriptions for video streaming over multiple paths. In *Proc. ACM Multimedia*, pp. 422–431, Berkeley, CA, Nov. 2003.
[5] T. Clausen and P. Jacquet. Optimized Link State Routing Protocol, Oct. 2003. IETF RFC 3626.
[6] D. Eppstein. Finding the k shortest paths. *SIAM J. on Comput.*, 28(2):652–673, August 1999.
[7] M.R. Garey and D.S. Johnson. *Computers and Intractability – A Guide to the Theory of NP-Completeness*. New York,NY: W.H. Freeman, 1979.
[8] D.B. Johnson, D.A. Maltz, and Y.-C. Hu. The dynamic source routing protocol for mobile ad hoc networks (DSR), July 2004. IETF Internet Draft, draft-ietf-manet-dsr-10.txt.
[9] S. Mao, Y.T. Hou, X. Cheng, H. D. Sherali, and S.F. Midkiff, Multipath routing for multiple description video in wireless ad hoc networks. In *Proc. IEEE INFOCOM,* pp.740-750, March 2005, Miami, FL.
[10] S. Mao, S. Lin, S.S. Panwar, Y. Wang, and E. Celebi. Video transport over ad hoc networks: Multistream coding with multipath transport. *IEEE J. Select. Areas Commun.*, 12(10):1721–1737, Dec. 2003.
[11] G.L. Nemhauser and L.A. Wolsey. Integer and Combinatorial Optimization. John Wiley & Sons, New York, NY, 1999.
[12] C.E. Perkins, E.M. Belding-Royer and S.R. Das, Ad hoc On-Demand Distance Vector (AODV) Routing, IETF RFC 3561, July 2003.
[13] H.D. Sherali and W.P. Adams. *A Reformulation-Linearization Technique for Solving Discrete and Continuous Nonconvex Problems*. Boston, MA: Kluwer Academic Publisher, 1999.
[14] H.D. Sherali, K. Ozbay, and S. Subramanian. The time-dependent shortest pair of disjoint paths problem: Complexity, models, and algorithms. *Networks*, 31(4):259–272, Dec. 1998.
[15] Y. Wang, R.R. Reibman, and S. Lin. Multiple description coding for video delivery. *Proc. IEEE*, 93(1):57–70, Jan. 2005.