

# Path Selection and Rate Allocation for Video Streaming in Multihop Wireless Networks

Sastry Kompella\*

Shiwen Mao<sup>†</sup>

Y. Thomas Hou<sup>†</sup>

Hanif D. Sherali<sup>‡</sup>

\* Information Technology Division, Naval Research Laboratory, Washington, DC

<sup>†</sup> The Bradley Department of Electrical and Computer Engineering, Virginia Tech, Blacksburg, VA

<sup>‡</sup> The Grado Department of Industrial and Systems Engineering, Virginia Tech, Blacksburg, VA

**Abstract**—In this paper, we investigate the important problem of optimal path selection and rate allocation for concurrent video sessions in a multihop wireless network. We present a novel formulation of the problem, which optimizes the application level performance such as video distortion, while seamlessly incorporating the network layer parameters such as packet delay and loss. For the formulated problem, we propose a *branch-and-bound* framework, predicated on the powerful *reformulation-linearization technique*, that can produce  $(1-\epsilon)$  optimal solutions. We demonstrate through analytical and simulation results that the proposed mechanism can fully utilize the network capacity and improve the quality of reconstructed video. The proposed approach provides an important methodology for addressing non-convex, non-polynomial programming problems that arise in wireless networks, especially those involving cross-layer design and optimizations.

**Index Terms**—Nonlinear programming, optimization, cross-layer design, ad hoc networks, video communications, path selection, rate allocation.

## I. INTRODUCTION

Recently, there has been considerable interest in supporting video communications in wireless ad hoc networks [6], [8], [11], [17]. The main technical challenges stem from the lack of infrastructure support, node mobility, and unreliable wireless communications. Among various mechanisms, path selection is arguably one of the most important for supporting video sessions. This is because the quality of a received video is highly dependent on the quality of the path(s) in terms of loss, delay, and delay variations. An efficient path selection algorithm should choose high quality paths for a video session, and would be especially appealing if it makes routing decisions directly based on the application layer performance such as video distortion.

For designing such a *cross-layer* strategy, this paper aims to address the following important problems: (i) given a set of available paths, which one(s) should be used, (ii) at what rate should the video be encoded and transmitted, and (iii) how to partition the rate among the chosen paths, each consisting of a sequence of wireless links with diverse qualities in terms of loss rate and available bandwidth such that the reconstructed video quality is maximized. These three questions are tightly coupled: the optimal rate is determined by the path selection and traffic proportioning strategy, and *vice versa*. Furthermore,

The work of S. Kompella was performed while he was a Ph.D student at the Bradley Department of Electrical and Computer Engineering, Virginia Tech, Blacksburg, VA.

the optimal rate vectors for all the concurrent video sessions are also closely dependent on each other: changing the rate allocation of one session may degrade the optimality of the rate vectors for other sessions. The interactions of competing video sessions in resource sharing must be considered, and an efficient strategy should take these factors into consideration in a holistic manner.

We formulate the path selection and rate allocation problem, which optimizes video application performance (i.e., distortion) by seamlessly incorporating the network layer parameters. That is, we take a *cross-layer* approach by modeling the application layer performance metric, video distortion, as a function of network layer behavior (i.e., path selection and traffic proportioning). For real-time applications, each data packet is associated with a *decoding deadline*, before which the packet must be successfully delivered in order to contribute to the reconstructed video quality. The impact of congestion (i.e., packet delay distribution) should be taken into consideration, in addition to that of link failures. In our formulation, interactions among competing video sessions contribute to the delays on shared links, and the end-to-end delay distribution is derived by applying the *Chernoff bound* approximation [3]. Such a formulation provides a guideline on choosing a set of optimal paths and rate vectors that provide the best video quality.

This problem formulation is a non-polynomial programming problem with a complex objective function and constraints. It does not possess such nice property as convexity. Each session's distortion is a function of the rates of all other sessions. Since non-polynomial programming problems are NP-hard in general, and our problem does not appear to possess any simplifying special structure, it is likely also NP-hard. Although this class of problems can be solved using metaheuristic algorithms (e.g., Genetic Algorithms [1]) for near-optimal solutions, we pursue to provide guarantees on the optimality gap of such near-optimal solutions. *The main contribution* of this paper is a specialized branch-and-bound-based framework, predicated on the powerful Reformulation-Linearization Technique (RLT), that can produce  $(1-\epsilon)$  optimal solutions to the problem of optimal path selection and rate allocation for concurrent video sessions. The proposed solution procedure is computationally efficient and provides an elegant trade-off between optimality and computation complexity.

The remainder of this paper is organized as follows. In Section II, we formulate the problem of optimal path selection

and rate allocation for concurrent video sessions. We then propose a branch-and-bound and RLT-based solution procedure in Section III. In Section IV, simulation results are presented to demonstrate the efficacy of the proposed approach. We discuss related work in Section V. Section VI concludes this paper.

## II. PROBLEM FORMULATION

We model a wireless ad hoc network as a directed graph  $\mathcal{G}(\mathcal{N}, \mathcal{L})$ , where  $\mathcal{N}$  is the set of vertices, representing mobile nodes, and  $\mathcal{L}$  is the set of wireless links in the network. We focus on link-layer metrics, assuming that lower layer dynamics can be translated into these link-layer metrics. For instance, we characterize each wireless (directed) link  $\{i, j\}$  by the following parameters: (i)  $c_{ij}$ : the capacity, or available bandwidth of link  $\{i, j\}$ ; and (ii)  $p_{ij}$ : the mean packet loss probability of link  $\{i, j\}$  due to transmission errors or link failures. In practice, these parameters can be measured by the nodes and distributed to the network [4], [15].

Consider a set of concurrent video sessions, denoted as  $\mathcal{S}$ , sustained in this network. Each video session  $\sigma \in \mathcal{S}$  has a source node  $z_\sigma$  and a destination node  $d_\sigma$ . For each session (i.e., a source-destination pair)  $z_\sigma-d_\sigma$ , there is a set of given paths, denoted by  $\mathcal{P}_\sigma$ . In practice, these paths can be precomputed by proactive routing protocols [4] or, by a reactive routing protocol [15]. The total rate of a video stream,  $R_\sigma$ , originated at source node  $z_\sigma$ , is bounded by  $\underline{R}_\sigma \leq R_\sigma \leq \overline{R}_\sigma$ ,  $\sigma \in \mathcal{S}$ , while the lower and upper bounds of  $R_\sigma$  are determined by the specific video encoder and the video sequence used by source node  $z_\sigma$ . This rate  $R_\sigma$  is to be allocated among all the paths in  $\mathcal{P}_\sigma$ .<sup>1</sup> Letting an element in the rate vector be  $R_\sigma^n$ ,  $n \in \mathcal{P}_\sigma$ , the following conditions must be satisfied.

$$\sum_{n \in \mathcal{P}_\sigma} R_\sigma^n = R_\sigma, \text{ and } R_\sigma^n > 0, \forall n \in \mathcal{P}_\sigma, \forall \sigma \in \mathcal{S}. \quad (1)$$

The main notation used in the paper is summarized in Table I.

### A. Link and Path Statistics

We derive link and path statistics in this section. These statistics will be used to compute the application layer video distortion in Section II-B.

1) *Load and delay on a link*: Based on the analysis provided in [12], the average rate of the aggregate traffic on link  $\{i, j\} \in \mathcal{L}$  is

$$\lambda_{ij} = \sum_{\sigma \in \mathcal{S}} \sum_{n \in \mathcal{P}_\sigma} \left[ \prod_{\{l, m\} \in \overline{\mathcal{P}}_\sigma^{n, ij}} (1 - p_{lm}) \right] \cdot R_\sigma^n. \quad (2)$$

Similarly, we could model the queueing delay on link  $\{i, j\} \in \mathcal{L}$  via an exponential distribution, i.e.,  $f_{ij}(y) = \alpha_{ij} \cdot e^{-\alpha_{ij}y}$ , for  $y \geq 0$ , where  $\alpha_{ij} = (c_{ij} - \lambda_{ij})$  is the residual bandwidth on the link.

<sup>1</sup>A path in  $\mathcal{P}_\sigma$  will be assigned a rate of zero if it is not selected. That is, path selection is made when the rate vectors are determined.

TABLE I  
NOTATION

Symbols	Definitions
$\{i, j\}$	A wireless link from node $i$ to node $j$ .
$\lambda_{ij}$	Average aggregate traffic on link $\{i, j\}$ .
$t_{ij}$	Delay on link $\{i, j\}$ .
$f_{ij}(y)$	Probability density function of $t_{ij}$ .
$\mathcal{S}$	Set of video sessions.
$\mathcal{P}_\sigma$	Path set of session $\sigma$ , from $z_\sigma$ to $d_\sigma$ .
$\mathcal{P}_\sigma^n$	A path in the set $\mathcal{P}_\sigma$ , from $z_\sigma$ to $d_\sigma$ .
$\Delta_\sigma$	Decoding deadline of session $\sigma$ .
$T_\sigma^n$	End-to-end delay on path $\mathcal{P}_\sigma^n \in \mathcal{P}_\sigma$ .
$T_\sigma$	Average end-to-end delay for session $\sigma$ .
$p_\sigma^n$	End-to-end loss rate of $\mathcal{P}_\sigma^n \in \mathcal{P}_\sigma$ .
$R_\sigma^n$	Rate of video session $\sigma$ on path $\mathcal{P}_\sigma^n \in \mathcal{P}_\sigma$ .
$R_\sigma$	Rate of video session $\sigma$ .
$\overline{R}_\sigma$	The maximum rate of video session $\sigma$ .
$\underline{R}_\sigma$	The minimum rate of video session $\sigma$ .
$D_\sigma^e$	End-to-end distortion of session $\sigma$ .
$D_\sigma^{enc}$	Encoding distortion of session $\sigma$ .
$D_\sigma^{cg}$	Distortion caused by congestion of session $\sigma$ .
$D_\sigma^{loss}$	Distortion caused by packet losses of session $\sigma$ .

2) *Path delay*: Recall that for each session  $\sigma \in \mathcal{S}$ , there is a set of paths  $\mathcal{P}_\sigma$  between the source node  $z_\sigma$  and the destination node  $d_\sigma$ . Let  $T_\sigma^n$  denote the delay on path  $\mathcal{P}_\sigma^n$  and  $T_\sigma$  the weighted average end-to-end delay among all the paths for session  $\sigma$ , then

$$T_\sigma = \sum_{n \in \mathcal{P}_\sigma} \frac{R_\sigma^n}{R_\sigma} \cdot T_\sigma^n. \quad (3)$$

Since the delay on path  $\mathcal{P}_\sigma^n$  consists of delays on all the links along the path, we have that  $T_\sigma^n = \sum_{\{i, j\} \in \mathcal{P}_\sigma^n} t_{ij}$ .

For the end-to-end delay, we can apply the *Chernoff Bound* [3] to obtain a good approximation, which is known to be accurate and computationally efficient [5]. Applying the Chernoff Bound, the distribution of  $T_\sigma^n$  can be approximated as [12]:

$$Pr\{T_\sigma^n \geq \Delta_\sigma\} \approx \left( \frac{\exp\{-F_\sigma^n(s_{\sigma, n}^*)\}}{s_{\sigma, n}^* \delta_{\sigma, n}(s_{\sigma, n}^*) \sqrt{2\pi}} \right), \quad (4)$$

where  $\delta_{\sigma, n}(s) = \sqrt{\frac{\partial^2 \log M_\sigma^n}{\partial s^2}}$ .

3) *End-to-end loss rate*: Assuming that the packet loss processes on the links are independent, the end-to-end loss probability for the path  $\mathcal{P}_\sigma^n \in \mathcal{P}_\sigma$  can be approximated as

$$p_\sigma^n = 1 - \prod_{\{i, j\} \in \mathcal{P}_\sigma^n} (1 - p_{ij}), \quad \forall n \in \mathcal{P}_\sigma, \forall \sigma \in \mathcal{S}. \quad (5)$$

### B. Video Distortion

In [20], Stuhlmüller *et al.* developed an empirical rate-distortion model for a hybrid motion compensated video encoder. For a video sequence encoded at a target coding rate  $R_\sigma$ , the average end-to-end distortion  $D_\sigma^e$  consists of the encoding distortion at the encoder ( $D_\sigma^{enc}$ ), the distortion due to congestion ( $D_\sigma^{cg}$ ), and the distortion due to packet loss ( $D_\sigma^{loss}$ ). That is,  $D_\sigma^e = D_\sigma^{enc} + D_\sigma^{cg} + D_\sigma^{loss}$ .

With our results on link and path statistics in Section II-A, we have

$$\begin{aligned}
D_\sigma^e &= \underbrace{D_0 + \frac{\omega}{R_\sigma - R_0}}_{D_\sigma^{enc}} + \underbrace{\kappa(1 - p_\sigma)Pr(T_\sigma > \Delta_\sigma)}_{D_\sigma^{cg}} + \underbrace{\kappa p_\sigma}_{D_\sigma^{loss}} \\
&= D_0 + \frac{\omega}{R_\sigma - R_0} + \\
&\quad \kappa \sum_{n \in \mathcal{P}_\sigma} \frac{R_\sigma^n}{R_\sigma} \{p_\sigma^n + (1 - p_\sigma^n)Pr(T_\sigma^n > \Delta_\sigma)\}, \quad (6)
\end{aligned}$$

where  $D_0$ ,  $\omega$ ,  $R_0$ , and  $\kappa$  are constants for a specific video codec and video sequence and can be obtained as discussed in [12]. Since this model takes into account the effects of INTRA coding and spatial loop filtering, it provides accurate estimates for end-to-end distortion [20].

### C. The Optimal Path Selection and Rate Allocation Problem

We are now ready to formulate the problem of optimal path selection and rate allocation for concurrent video sessions (OPT-PSRA), with the objective of minimizing application layer video distortion. Mathematically, Problem OPT-PSRA can be stated as in Eqs. (7)–(12).

In Problem OPT-PSRA, the objective function (7) is the sum of the average distortion of all the concurrent video sessions. The goal is to obtain the best possible rate vectors that would minimize (7) over a given set of paths for each video session. It should be noted that alternative objective functions, such as  $\max\{D_\sigma^e\}$  or an objective function in the form of a utility function  $\sum_\sigma f(D_\sigma^e)$  (e.g., a logarithmic utility function), can also be handled using the same solution procedure with minor modifications.

Observe that the objective function (7) and the constraints (11) and (12) are non-polynomial, non-convex functions of both  $\{R_\sigma\}_{\sigma \in \mathcal{S}}$  and  $\{R_\sigma^n\}_{n \in \mathcal{P}_\sigma}$ . The rates of all the video sessions are closely coupled in (7). Since non-polynomial problems are NP-hard in general, and Problem OPT-PSRA does not appear to possess any simplifying special structure, it is likely also NP-hard.

In the following section, we present a branch-and-bound solution procedure for Problem OPT-PSRA, predicated on the powerful RLT approach. Our proposed solution procedure can produce a solution within a relative error of  $\epsilon$  to the global optimum ( $\epsilon \equiv 0$  if exact optimum is desired).

## III. SOLUTION PROCEDURE

Our solution procedure is based on embedding a novel relaxation technique called the *Reformulation-Linearization Technique* (RLT) in a *branch-and-bound* framework. In this section, we first reformulate the non-polynomial terms in Problem OPT-PSRA, so as to convert it to a polynomial, non-linear optimization problem. Then, we present details of embedding the RLT-relaxation of this polynomial problem into a branch-and-bound framework, and describe the solution procedure that produces  $(1-\epsilon)$  optimal solutions.

### A. Reformulation

Due to the existence of non-polynomial terms in Problem OPT-PSRA, our first goal is to reformulate this problem into a *polynomial programming problem*, which will simplify the objective function as well as the constraints.

In the objective function (7), there are three sets of non-polynomial terms. In order to transform the first two non-polynomial terms, we define new variables  $u_\sigma = 1/(R_\sigma - R_0)$  and  $w_\sigma = (1/R_\sigma) \cdot \sum_{n \in \mathcal{P}_\sigma} R_\sigma^n \cdot p_\sigma^n$ . Substituting  $u_\sigma$  and  $w_\sigma$  into the objective function, we get two linear terms  $\omega \cdot u_\sigma$  and  $\kappa \cdot w_\sigma$ , respectively, and two sets of new polynomial constraints  $u_\sigma \cdot (R_\sigma - R_0) = 1$  and  $w_\sigma \cdot R_\sigma = \sum_{n \in \mathcal{P}_\sigma} R_\sigma^n \cdot p_\sigma^n$ .

In order to transform the third non-polynomial term in Eq. (7), i.e., the product of fractions, recall that  $\alpha_{ij}$  denotes the available bandwidth on a link  $\{i, j\}$ , i.e.,

$$\alpha_{ij} = c_{ij} - \sum_{\sigma \in \mathcal{S}} \sum_{n \in \mathcal{P}_\sigma} \theta_{ij}^{\sigma, n} \cdot R_\sigma^n, \quad (13)$$

where  $\theta_{ij}^{\sigma, n} = \prod_{\{l, m\} \in \bar{\mathcal{P}}_\sigma^{n, ij}} (1 - p_{lm})$ . Note that  $\theta_{ij}^{\sigma, n}$  is a constant, since the paths are given. Let  $g_\sigma^n$  denote the weighted packet overdue probability on path  $\mathcal{P}_\sigma^n$ , i.e.,  $g_\sigma^n = \frac{R_\sigma^n}{R_\sigma} \cdot Pr(T_\sigma^n > \Delta_\sigma)$ . Again, define a substitution variable  $v_{ij} = \frac{1}{\alpha_{ij} - s_{\sigma, n}^*}$  to convert the fractions in  $Pr(T_\sigma^n > \Delta_\sigma)$  to polynomial form, i.e.,

$$g_\sigma^n = \frac{R_\sigma^n}{R_\sigma} \cdot \left[ \frac{e^{-s_{\sigma, n}^* \Delta_\sigma}}{s_{\sigma, n}^* \delta_{\sigma, n} \sqrt{2\pi}} \prod_{\{i, j\} \in \mathcal{P}_\sigma^n} \alpha_{ij} \cdot v_{ij} \right]. \quad (14)$$

This product form motivates us to apply logarithms on both sides of Eq. (14), which will lead to a linear constraint. This way, the complexity on the objective function can be effectively moved into the constraints.

Once the objective function is simplified, we now deal with the constraints of the form  $y = \log(x)$ . We can linearize this logarithmic relationship over some proper tight bounds using a *polyhedral outer approximation* comprised of a *convex envelope* (a chord connecting the end points in concert with several tangential supports). For instance, if  $x$  is bounded as  $0 < x_0 \leq x \leq 1$ , these constraints can be written as follows.

$$\begin{cases} y \geq \frac{\log(x_0)}{1-x_0} \cdot (1-x) \\ y \leq \log(x_k) + \frac{x-x_k}{x_k}, \quad k = 1, \dots, k_{max}, \end{cases} \quad (15)$$

where  $x_k = x_0 + (1-x_0) \cdot \frac{k-1}{k_{max}-1}$ , for  $k = 1, \dots, k_{max}$ . A four-point tangential approximation can be obtained by letting  $k_{max} = 4$ . As a result, every logarithmic relationship translates to five linear constraints.

With the above re-formulation, we can now rewrite Problem OPT-PSRA as a *polynomial programming problem* ( $p$ -PSRA). Problem OPT-PSRA is now transformed into a polynomial non-linear programming problem (NLP) of order two. The highly complex objective function (7) is greatly simplified (i.e., linearized) and the complexity is shifted into the constraints in the polynomial form. In the remainder of this section, we present our branch-and-bound and RLT-based solution procedure to solve Problem OPT-PSRA.

## OPT-PSRA

Minimize

$$D = \sum_{\sigma \in \mathcal{S}} \left\{ D_0 + \frac{\omega}{R_\sigma - R_0} + \kappa \sum_{n \in \mathcal{P}_\sigma} \frac{R_\sigma^n}{R_\sigma} \left\{ p_\sigma^n + (1 - p_\sigma^n) \left\{ \frac{e^{-s_{\sigma,n}^* \Delta_\sigma}}{s_{\sigma,n}^* \delta_{\sigma,n}(s_{\sigma,n}^*) \sqrt{2\pi}} \right. \right. \right. \\ \left. \left. \left. \prod_{\{i,j\} \in \mathcal{P}_\sigma^n} \frac{c_{ij} - \sum_{\varphi \in \mathcal{S}} \sum_{k \in \mathcal{P}_\varphi} \left[ \prod_{\{l,m\} \in \bar{\mathcal{P}}_\varphi^{k,ij}} (1 - p_{lm}) \right] R_\varphi^k}{c_{ij} - \sum_{\varphi \in \mathcal{S}} \sum_{k \in \mathcal{P}_\varphi} \left[ \prod_{\{l,m\} \in \bar{\mathcal{P}}_\varphi^{k,ij}} (1 - p_{lm}) \right] R_\varphi^k - s_{\sigma,n}^*} \right\} \right\} \right\} \quad (7)$$

subject to

$$R_\sigma = \sum_{n \in \mathcal{P}_\sigma} R_\sigma^n, \quad \forall \sigma \in \mathcal{S} \quad (8)$$

$$\underline{R}_\sigma \leq R_\sigma \leq \bar{R}_\sigma, \quad \forall \sigma \in \mathcal{S} \quad (9)$$

$$\sum_{\sigma \in \mathcal{S}} \sum_{n \in \mathcal{P}_\sigma} \left[ \prod_{\{l,m\} \in \bar{\mathcal{P}}_\sigma^{n,ij}} (1 - p_{lm}) \right] \cdot R_\sigma^n \leq (1 - \epsilon) \cdot c_{ij}, \quad \forall \{i,j\} \in \mathcal{P}_\sigma \quad (10)$$

$$\delta_{\sigma,n}^2 = \sum_{\{i,j\} \in \mathcal{P}_\sigma^n} \frac{1}{(c_{ij} - \sum_{\varphi \in \mathcal{S}} \sum_{k \in \mathcal{P}_\varphi} \left[ \prod_{\{l,m\} \in \bar{\mathcal{P}}_\varphi^{k,ij}} (1 - p_{lm}) \right] R_\varphi^k - s_{\sigma,n}^*)^2}, \quad \forall n \in \mathcal{P}_\sigma, \forall \sigma \in \mathcal{S} \quad (11)$$

$$\Delta_\sigma = \sum_{\{i,j\} \in \mathcal{P}_\sigma^n} \frac{1}{c_{ij} - \sum_{\varphi \in \mathcal{S}} \sum_{k \in \mathcal{P}_\varphi} \left[ \prod_{\{l,m\} \in \bar{\mathcal{P}}_\varphi^{k,ij}} (1 - p_{lm}) \right] R_\varphi^k - s_{\sigma,n}^*}, \quad \forall n \in \mathcal{P}_\sigma, \forall \sigma \in \mathcal{S}. \quad (12)$$

## B. Branch-and-Bound

Branch-and-bound is an iterative relaxation algorithm [18] that seeks to obtain a  $(1-\epsilon)$  optimal solution for a nonlinear problem (NLP). Here  $\epsilon \geq 0$  is a constant reflecting some desired optimality tolerance. In 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 choose a novel relaxation technique called the *Reformulation-Linearization Technique* (RLT) [18], [19] to obtain a linear programming (LP) relaxation (see Section III-C). 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 NLP problem, a *local search algorithm*, as explained in Section III-D must 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.

The branch-and-bound procedure is based on the idea of *divide-and-conquer*. That is, the original problem,  $P$ , is partitioned into sub-problems, each having a smaller feasible solution space, based on the solution provided by the LP relaxation. This branching process is carried out recursively to obtain two new sub-problems at each node of the *branch-and-bound tree*. The partitioning of the original solution space, i.e., the *branching rule*, will be explained further in Section III-C. The sub-problems are 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  $P$ . At any given point,

the lower bound and upper bound for  $P$  are computed as

$$\begin{cases} LB = \min\{LB_k : \text{Problem } k \in L\} \\ UB = \min\{UB_k : \text{Problem } k \text{ explored thus far}\}. \end{cases} \quad (16)$$

The method proceeds by choosing the next problem to partition from the problem list. In our approach, the problem  $k$  having the worst (or smallest)  $LB$  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  $P$  are updated as in Eq. (16). At any given iteration, if  $LB \geq (1-\epsilon) \cdot UB$ , the procedure exits with a  $(1-\epsilon)$  optimal solution. Otherwise, for any problem  $k$  in the problem list, if  $LB_k \geq (1-\epsilon) \cdot UB$ , no globally optimal solution that improves beyond the  $\epsilon$ -tolerance can exist in the subspace of the feasible region represented by this node. Therefore, this node can be removed from the branch-and-bound tree. In this manner, the branch-and-bound can *prune* or *fathom* certain branches of the branch-and-bound tree, without solving all the nodes in the branch to completion.

## C. The Reformulation-Linearization Technique

RLT is a relaxation technique that can be used to produce tight *polyhedral outer approximations* or *linear programming relaxations* for an underlying nonlinear, nonconvex polynomial programming problem, which, in essence, can provide a tight lower bound on a minimization problem [18], [19]. In our approach, the key to obtaining a  $(1-\epsilon)$  optimal solution is to embed RLT into the branch-and-bound framework as discussed earlier, and the proposed method is known to converge to a global optimum.

In the RLT procedure, nonlinear implied constraints are generated by taking the products of bounding terms of the

decision variables, up to a suitable order and also, possibly products of other defining constraints of the problem. The resulting problem is subsequently linearized by variable substitutions, one for each nonlinear term appearing in the problem, including both the objective function and the constraints.

For instance, the second order term  $u_\sigma \cdot R_\sigma$  in Problem  $p$ -PSRA can be viewed as a single term, for which we can introduce a new variable  $\mu_\sigma$ , thereby substituting  $\mu_\sigma = u_\sigma \cdot R_\sigma$ . Since  $u_\sigma$  and  $R_\sigma$  are each bounded by  $(u_\sigma)_L \leq u_\sigma \leq (u_\sigma)_U$  and  $(R_\sigma)_L \leq R_\sigma \leq (R_\sigma)_U$ , respectively, we generate the following relational constraints, which are known as *RLT bound-factor product constraints*.

$$\begin{cases} \{[u_\sigma - (u_\sigma)_L] \cdot [R_\sigma - (R_\sigma)_L]\}_L \geq 0 \\ \{[u_\sigma - (u_\sigma)_L] \cdot [(R_\sigma)_U - R_\sigma]\}_L \geq 0 \\ \{[(u_\sigma)_U - u_\sigma] \cdot [R_\sigma - (R_\sigma)_L]\}_L \geq 0 \\ \{[(u_\sigma)_U - u_\sigma] \cdot [(R_\sigma)_U - R_\sigma]\}_L \geq 0, \end{cases} \quad (17)$$

where  $\{\cdot\}_L$  denotes a linearization step under the substitution  $\mu_\sigma = u_\sigma \cdot R_\sigma$ . From the above relationships and by substituting  $\mu_\sigma = u_\sigma \cdot R_\sigma$ , we can generate linear RLT constraints for  $\mu_\sigma$ . We therefore replace the second-order term  $u_\sigma \cdot R_\sigma$  with the linear term  $\mu_\sigma$ , wherever it appears in  $p$ -PSRA, and introduce the above linear bound-factor RLT constraints for  $\mu_\sigma$  into the Problem  $p$ -PSRA formulation. Similarly, we define new variables for all the remaining nonlinear terms in Problem  $p$ -PSRA, and make substitutions in the same manner.

It is worth noting that, when incorporating RLT into the branch-and-bound framework, the original bounding set  $\Omega$  is decomposed into two corresponding hyper-rectangles, based on a so called *branching variable*. In RLT, the discrepancy between an RLT variable and the corresponding nonlinear term that the RLT variable represents is called the *relaxation error*, and we choose a branching variable that yields the largest relaxation error. Such a branching rule ensures that all the discrepancies will be driven to zero from iteration to iteration. After replacing all non-linear terms as above and adding the corresponding RLT constraints into the Problem  $p$ -PSRA formulation, we obtain a *linear programming relaxation problem* ( $\ell$ -PSRA), for which many efficient (polynomial-time) solution techniques and tools are available.

#### D. The Local Search Algorithm

As discussed in Section III-B, in the branch-and-bound procedure, the solution to the relaxation problem is usually infeasible to the original problem. This problem can be resolved by finding a feasible solution to the original problem via a local search algorithm that starts from the infeasible solution. Let  $\mathbf{R}$  and  $\mathbf{R}^n$  be vectors having components  $R_\sigma$  and  $R_\sigma^n$ , respectively. For Problem OPT-PSRA, we adopt the following local search strategy that computes a feasible solution  $(\hat{\mathbf{R}}, \hat{\mathbf{u}}, \hat{\mathbf{v}}, \hat{\alpha}, \hat{\delta}, \hat{\mathbf{s}})$  from the solution to the relaxation problem  $(\mathbf{R}, \mathbf{u}, \mathbf{v}, \alpha, \delta, \mathbf{s})$ . Specifically, since the rates of the video sessions obtained from the solution to the relaxation problem are always feasible to the original problem (i.e., the stability constraints are always satisfied and the rates are always within the lower and upper bounds), we have that  $\mathbf{R} = \hat{\mathbf{R}}$ . Using the value of  $\hat{\mathbf{R}}$  we can compute a feasible

TABLE II  
PERFORMANCE OF THE PROPOSED ALGORITHM FOR VARIOUS INSTANCES OF PROBLEM OPT-PSRA

Size	Sessions	Paths/Sess.	$\epsilon$	Init. Solution	(1- $\epsilon$ ) optimal
30	2	1	0.05	106.64	106.64
30	3	1	0.05	181.85	179.26
30	4	1	0.05	249.09	240.66
50	4	1	0.05	255.59	252.13
50	6	1	0.1	419.44	366.41
50	8	1	0.1	537.65	500.33
50	10	1	0.1	665.39	640.10
50	3	2	0.1	200.45	190.67
50	4	2	0.1	295.01	282.60
50	5	2	0.1	369.19	301.53

solution to the original problem OPT-PSRA,  $(\mathbf{R}, \mathbf{u}, \mathbf{v}, \alpha, \delta, \mathbf{s})$ , based on the relationships explained earlier in Section III-A.

#### IV. SIMULATION STUDIES

In this section, we present simulation results for Problem OPT-PSRA. In each simulation, a wireless ad hoc network is generated by placing a number of nodes at random locations in a rectangular region. A wireless link exists if a node is within the radio range of a transmitting node. As discussed, a set of preselected paths are precomputed using a  $k$ -disjoint path routing algorithm for each source-destination pair, which are randomly chosen from the set of nodes  $\mathcal{N}$ . In the simulations, each video session has a rate bounded by 20 Kb/s and 200 Kb/s. We used an H.263+ codec and the first 200 frame of the ‘‘Foreman’’ trace in the quarter common intermediate format (QCIF). The video was encoded at 12.5 frames per second and an intra rate of 1/9. The rate-distortion parameters are obtained from [20]. Failure probabilities of the wireless links are chosen from a uniform distribution between [1%, 5%]; the bandwidth of a link is chosen from a uniform distribution between [50 Kb/s, 400 Kb/s]. The proposed solution procedure is implemented in C, and the LINDO API 3.0 is used for solving the LP relaxation Problem  $\ell$ -PSRA.

##### A. Performance on Different Instances of Problem OPT-PSRA

We first examine the performance of the proposed solution procedure with different instances of Problem OPT-PSRA, which are presented in Table II. In each of these case studies, we examine the performance of the proposed solution procedure in the presence of multiple shared links and bottlenecks. For example, the last three cases in Table II are for the same 50-node network, with three, four, and five sessions, respectively, and two paths for each session. In these case studies, the proposed algorithm performs both rate allocation and path selection (i.e., proportional routing). The decoding deadline is 0.2 s for all the cases in Table II.

The fifth column of Table II presents the total distortion values found by solving the corresponding  $UB$  for Problem 1 (i.e., the first node in the tree) in the RLT-based branch-and-bound algorithm; the sixth column presents the  $(1-\epsilon)$  optimal solution found by the algorithm. We find that the corresponding values between these two columns are very close to each other. This clearly demonstrates that the polyhedral outer

approximation and the RLT-based LP relaxations used in the solution procedure are well designed and tight.

### B. Comparison with Network-centric Scheme

In the remainder of this section, we compare the performance of the proposed approach with a network-centric rate allocation scheme. Specifically, a *max-min* rate allocation scheme (called Max-Min throughout this paper) is implemented, which is widely regarded as a fair rate allocation policy [2]. In Max-Min, fairness is achieved by maximizing the minimum rate allocation in the network without exceeding its upper bound and the capacity of each link. Note that the max-min rate allocation is based on the fluid flow model and does not explicitly consider queuing delay at the links. Therefore, we need to compute the max-min fair rate allocations for a prescribed link utilization factor, which specifies the maximum percentage of capacity that can be used on a given link. Using this scheme, we find the rate allocation for each video session and then compute the total distortion using Eq. (7).

In Figure 1, we plot the average distortions found by the proposed algorithm and Max-Min for various decoding deadlines. The network consists of 50 nodes with 5 concurrent video sessions. There are two paths available for each session. To be fair in comparison, the Max-Min scheme is executed for various link utilization factors, ranging from 30% to 80%. We find that the average distortion quickly decreases when the delay constraint is relaxed, but more importantly, the branch-and-bound/RLT approach outperforms the Max-Min approach by a significant margin: there is a big gap between the branch-and-bound/RLT curve and the lower bounding envelope of the Max-Min curves. For example, when the decoding deadline is 0.2 s, the average distortion achieved by our approach is 60.31, which translates to a Peak-Signal-Noise-Ratio (PSNR) of 30.33 dB (computed as  $10 \cdot \log_{10}(255 * 255 / D_{\sigma}^e)$ ); the average distortion achieved by Max-Min when  $\rho=50\%$  is 107.61, which translates to a PSNR of 27.81 dB (the best among all the Max-Min curves). There is a 78.4% reduction in distortion and a 2.52 dB improvement in average PSNR. Note that such an improvement is significant in terms of perceived video quality, since usually a half dB difference in PSNR is noticeable.

We now simulate the actual video streaming in this 50-node network for the  $\Delta_{\sigma} = 0.2$  s case, using the “foreman” video trace. In the simulations, the source nodes transmit packetized video traffic along the paths according to the rate vectors computed, while the destination nodes reconstruct video frames from received packets and compute their PSNR values. For Session Two, under the branch-and-bound/RLT scheme, the average PSNR is 32.27 dB, while under Max-Min for link utilization of 30%, 50%, 70%, and 80%, the PSNRs are 28.73 dB, 30.20 dB, 29.62 dB, and 26.88 dB respectively. The proposed approach achieves a 2.07 dB gain in average PSNR over Max-Min with  $\rho=50\%$ , which is the best among all the Max-Min schemes. In Figure 2, we present Frame 160 obtained by various schemes. Again, we find that although the frame delivered by Max-Min with  $\rho=50\%$  is the best among all the Max-Min frames, it is still inferior to the frame obtained by our algorithm.

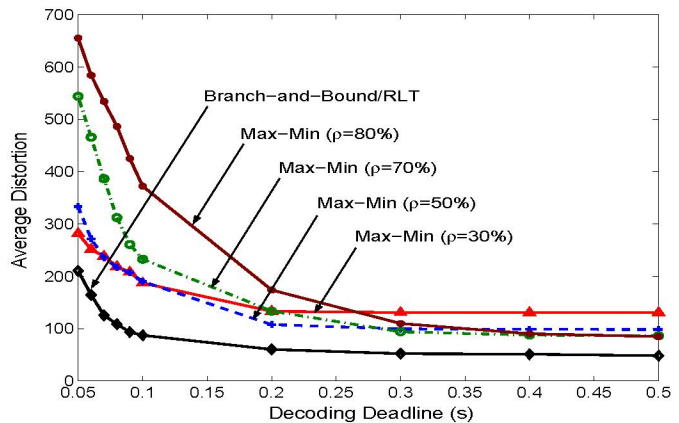


Fig. 1. Average distortion versus decoding deadline for a 50-node network with five sessions and two paths per session.

## V. RELATED WORK

The optimal traffic proportioning problem, i.e., given a set of calls, how to partition them to the set of given paths such that the overall throughput is maximized, has been studied extensively in the context of telephone networks (see [16] and references therein). Recently, a localized approach to the proportional routing problem is proposed in [14]. In addition, the problem of utility maximization with multi-path routing has been studied in the context of flow/congestion control for elastic data in a few recent works [9], [10]. In such problems, each session is associated with a concave, univariate utility function of its total rate.

These prior efforts motivate us to pursue this important problem, particularly when video quality is the optimization objective. The problem we study in this paper differs from those analyzed by prior works in that we need information on link metrics and on all the active video sessions. The formulation is considerably more complex in that we model the end-to-end delay distributions, which is required for real-time traffic with tight decoding deadlines. A session’s distortion is affected by other sessions in the network, making it impossible to break the original problem into simpler subproblems or to derive an easier-to-solve dual problem. As a result, prior approaches (developed for elastic data traffic based on flow models) could not be applied to our problem.

## VI. CONCLUSIONS

In this paper, we investigated the problem of path selection and rate allocation for concurrent video sessions in an ad hoc network. We formulated the problem as a nonconvex programming problem and developed an efficient solution procedure based on the branch-and-band framework and the Reformulation-Linearization Technique. This solution approach is shown to produce  $(1-\epsilon)$  optimal solutions, and provide guarantees on the optimality gap.

## ACKNOWLEDGEMENTS

The work of S. Kompella was supported by NSF IGERT program under Grant DGE-9987586. The work of Y.T. Hou



Fig. 2. Reconstructed Frame 160 from the “Foreman” sequence for Session Two.

has been supported in part by NSF under Grant CNS-0347390 and ONR Grant N00014-05-1-0481. The work of H.D. Serali has been supported by NSF under Grant 0552676.

#### REFERENCES

- [1] T. Back, D. Fogel, and Z. Michalewicz, Eds., *Handbook of Evolutionary Computation*, Oxford University Press, New York, NY, 1997.
- [2] D. Bertsekas and R. Gallager, *Data Networks*, Prentice Hall, second edition, 1991.
- [3] H. Chernoff, “A measure of asymptotic efficiency for tests of a hypothesis based on the sum of observations,” *Annals of Mathematical Statistics*, vol. 23, pp. 493–507, 1952.
- [4] T. Clausen and P. Jacquet, “Optimized Link State Routing Protocol,” Oct. 2003, IETF RFC 3626.
- [5] A. Elwalid, D. Heyman, T.V. Lakshman, D. Mitra, and A. Weiss, “Fundamental bounds and approximations for atm multiplexers with applications to video teleconferencing,” *IEEE J. Select. Areas Commun.*, vol. 13, no. 6, pp. 953–962, August 1995.
- [6] H. Gharavi and K. Ban, “Dynamic adjustment packet control for video communications over ad-hoc networks,” in *Proc. IEEE ICC 2004*, Paris, France, June 2004, pp. 3086–3090.
- [7] C. Goldie and C. Klüppelberg, *Subexponential Distributions*, pp. 435–459, Birkhäuser Publishing Ltd., Basel, Switzerland, 1998.
- [8] Y.-C. Hu and D.B. Johnson, “Design and demonstration of live audio and video over multihop wireless ad hoc networks,” in *Proc. IEEE MILCOM 2002*, Anaheim, CA, Oct. 2002, pp. 7–10.
- [9] F.P. Kelly, A.K. Maulloo, and D.K.H. Tan, “Rate control in communication networks: Shadow prices, proportional fairness and stability,” *J. Oper. Res. Soc.*, vol. 49, no. 3, pp. 237–252, Mar. 1998.
- [10] X. Lin and N.B. Shroff, “Utility maximization for communication networks with multipath routing,” *IEEE Trans. Automatic Control*, vol. 51, no. 5, pp. 766–781, May 2006.
- [11] S. Mao, Y.T. Hou, X. Cheng, H.D. Serali, and S.F. Midkiff, “Multi-path routing for multiple description video over wireless ad hoc networks,” in *Proc. IEEE INFOCOM 2005*, Miami, FL, Mar. 2005, pp. 740–750.
- [12] S. Mao, S. Kompella, Y.T. Hou, H.D. Serali and S.F. Midkiff, “Routing for multiple concurrent video sessions in wireless ad hoc networks,” *IEEE Trans. Vehicular Technology*, vol. 55, no. 1, pp. 317–327, Jan. 2006.
- [13] J. Moy, “OSPF version 2,” Apr. 1998, IETF RFC 2328.
- [14] S. Nelakuditi, Z.-L. Zhang, R.P. Tsang, and D.H.C. Du, “Adaptive proportional routing: A localized QoS routing approach,” *IEEE/ACM Trans. Networking*, vol. 10, no. 6, pp. 790–804, Dec. 2002.
- [15] C.E. Perkins, E.M. Royer, and S.R. Das, “Quality of service in ad hoc on-demand distance vector routing,” July 2000, IETF Internet Draft draft-ietf-manet-qos-00.txt.
- [16] K.W. Ross, *Multiservice loss models for broadband telecommunication networks*, Springer, 1995.
- [17] E. Setton, X. Zhu, and B. Girod, “Congestion-optimized multi-path streaming of video over ad hoc wireless networks,” in *Proc. ICME 2004*, Taipei, Taiwan, June 2004, pp. 1619–1622.
- [18] H.D. Serali and W.P. Adams, *A Reformulation-Linearization Technique for Solving Discrete and Continuous Nonconvex Problems*, Kluwer Academic Publisher, Boston, MA, 1999.
- [19] H.D. Serali and C.H. Tuncbilek, “A global optimization algorithm for polynomial programming problems using a reformulation-linearization technique,” *J. Global Optim.*, vol. 2, no. 1, pp. 101 – 112, 1992.
- [20] K. Stuhlmüller, N. Farberand, M. Link, and B. Girod, “Analysis of video transmission over lossy channels,” *IEEE J. Select. Areas Commun.*, vol. 18, no. 6, pp. 1012–1032, June 2000.