

Optimal Rate Control for Video Transport over Multi-hop Wireless Networks

Sastry Kompella[†] Shuwen Mao[†] Y. Thomas Hou[†] Hanif D. Sherali[‡]

[†]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

Email: {sastryk, smao, thou, hanifs}@vt.edu

Abstract—Video communication is an important application area for a multihop wireless network. This paper studies the problem of finding the optimal encoding rates for a number of video sessions in such network. The objective is to maximize the video quality at the receivers and the optimization space takes into consideration, the interaction among all the active video sessions. A branch-and-bound solution procedure is proposed to solve this nonconvex, non-polynomial programming problem. Using analytical and simulation results, we show that this solution procedure is an effective approach for addressing such complex cross-layer optimization problem.

I. INTRODUCTION

Video communications over multihop wireless networks has received a lot of attention in recent years. In wireless multihop networks, nodes cooperate with each other in relaying packets through the network. These networks can be deployed with minimal infrastructure, making them especially suitable for operating in special conditions. However, they are also prone to frequent link failures and topology changes, which makes it hard to support multimedia communications.

Multimedia communications, in particular video applications, are typically characterized by their sensitivity to delay and bandwidth requirements. For video applications, packets transmitted over a multihop wireless network need to meet a pre-specified deadline in order to be successfully decoded at the receiver. The decoded video quality is, therefore, affected by two factors: distortion introduced by the compression at the encoder, and distortion due to packet loss or delay, which is dependent on the rate at which video is encoded.

A naive approach to computing the video rate for each session, is to assign the bottleneck bandwidth of the path on which the session will be transmitted. However, this is not always optimal, because at maximum load, the delay on the link increases indefinitely. In a bandwidth limited environment, therefore, video applications are expected to achieve maximum decoded video quality for some intermediate rate. In general, optimizing individual network layer parameters such as available bottleneck bandwidth does not necessarily guarantee optimality. In this paper, therefore, we represent the tradeoff between rate and the final video quality as a function of multiple network layer metrics (e.g., bandwidth, delay). We consider the impact of congestion (modeled using packet delay distributions) in addition to link failures in computing optimal rates for encoding each session. We also consider the

interaction of competing video sessions as they share the same pool of network resources.

We consider the following problem: given a set of paths, one for each source destination pair, what is the optimal rate at which each should be encoded, such that the reconstructed video quality at the receiver is maximized. We show that the optimal rates for all the sessions are tightly coupled, i.e., changing the rate allocation of one session may degrade the quality of the other sessions. The objective function minimizes an application layer performance metric (i.e., average video distortion) as a function of network layer performance metrics (e.g., bandwidth, loss, and path correlation). Our formulation shows that this is a non-polynomial programming problem, with a complex objective function and constraints, where the distortion of each session is shown to be a function of the rates of all other sessions, and hence does not possess any special structure such as convexity.

One possible approach to solve this problem is to apply metaheuristics such as Genetic Algorithms (GA). Indeed, we studied such an approach in our prior work [9] and showed that GA could be effective in addressing such type of complex optimization problems. But, as with any other metaheuristics, GA is not able to offer a guaranteed performance bound, which is important in understanding the performance limit of the objective function. In this paper, we address this issue, using the so-called branch-and-bound framework [10]. Specifically, we show that, branch-and-bound framework, along with reformulation-linearization technique (RLT) [11], can produce ϵ -optimal solutions to our problem. The level of accuracy can be tuned based on tolerance in computational complexity.

The remainder of this paper is organized as follows. In Section II, we formulate the problem of optimal rate control for video transport. In Section III, we propose a branch-and-bound and RLT-based solution procedure. Section IV presents simulation results to illustrate the efficacy of the proposed approach. Section V discusses related work and Section VI concludes the paper.

II. PROBLEM FORMULATION

We model a wireless ad hoc network as a time-varying directed graph $\mathcal{G}(\mathcal{N}, \mathcal{L})$, where \mathcal{N} is the set of vertices, representing nodes, and \mathcal{L} is the set of wireless links in the network. We assume that a link exists between node i and node j if they are within transmission range of each other.

Each link $\{i, j\} \in \mathcal{L}$ in the wireless network is characterized by the capacity, or available bandwidth of link b_{ij} ; and the mean packet loss probability p_{ij} , due to node or link failures. In practice, these parameters can be measured by nodes and distributed to the network [4].

Within this network, we assume there exists a set of video sessions, denoted as \mathcal{F} , with each session $f \in \mathcal{F}$ having a source node S_f and a destination node D_f . We also assume that there is a given path connecting S_f and D_f , denoted by L_f . In practice, these paths can be precomputed by proactive routing protocols, or discovered by a reactive routing protocol. The rate of a video stream, R_f , is bounded as $\underline{R}_f \leq R_f \leq \overline{R}_f, \forall f \in \mathcal{F}$, where, the lower and upper bounds of R_f are determined by the specific video encoder and video sequence used at the source node S_f .

TABLE I
NOTATION

Symbol	Definition
$\mathcal{G}\{\mathcal{N}, \mathcal{L}\}$	Representation of the wireless mobile network.
\mathcal{N}	Set of vertices.
\mathcal{L}	Set of edges.
$\{i, j\}$	A wireless link from node i to node j .
b_{ij}	Bandwidth of link $\{i, j\}$.
p_{ij}	Packet loss probability of link $\{i, j\}$.
λ_{ij}	Average aggregate traffic on link $\{i, j\}$.
ρ_{ij}	Utilization of link $\{i, j\}$.
\mathcal{F}	Set of video sessions.
f	A video session in \mathcal{F} .
S_f	Source node of session f .
D_f	Destination node of session f .
L_f	Path connecting S_f to D_f .
Δ_f	Decoding deadline of session f .
T_f	End-to-end delay on path L_f .
p_f	End-to-end loss rate of L_f .
R_f	Rate of video session f on path L_f .
\bar{R}_f	The maximum rate of video session f .
\underline{R}_f	The minimum rate of video session f .
D_f^e	End-to-end distortion of session f .
D_f^{enc}	Encoding distortion of session f .
D_f^{cg}	Distortion caused by congestion of session f .
D_f^{loss}	Distortion caused by packet losses of session f .

The rate control problem for video sessions can be described as follows. Given a wireless ad hoc network $\mathcal{G}(\mathcal{N}, \mathcal{L})$, a set of video sessions \mathcal{F} with source-destination pairs $\{S_f, D_f\}$, $f \in \mathcal{F}$, and a feasible path L_f between each source-destination pair $\{S_f, D_f\}$, $f \in \mathcal{F}$, find a feasible set of rate vectors, one for each source destination pair, such that the total (or average) distortion of the video sessions is minimized. This problem OPT-ARC is formally stated as shown in (1)–(5) (shown in the next page).

We now provide an interpretation for the problem formulation. In the OPT-ARC problem, the objective function D in (1) is the sum of the average distortion D_f^e of all the concurrent video sessions. The relationship between the average distortion D_f^e and the link statistics for the path L_f is given by the rate-distortion model. A rate-distortion model for video coding describes the distortion achieved by the coder, as a function of the video rate. Using an empirical

rate-distortion model developed by Stuhlmüller et al, in [12], an expression for D_f^e was derived in [9]. In [9], we showed that the decoded video quality at the receiver is affected by three factors; encoding distortion caused by the lossy video coder, D_f^{enc} , distortion caused by overdue video packets (i.e., packets experiencing delay greater than the decode-deadline Δ_f , due to congestion in the network), D_f^{cg} , and the distortion caused by lost video packets (i.e., due to link failure or other transmission errors), D_f^{loss} . Here, D_0 , ω , R_0 , and κ are constants for a specific video coder (with fixed encoding parameters) and video sequence, which are estimated from empirical rate-distortion curves by training and curve matching [12]. Therefore,

$$D_f^e = D_f^{enc} + D_f^{cg} + D_f^{loss} \quad (6)$$

and,

$$D_f^e = \underbrace{D_0 + \frac{\omega}{R_f - R_0}}_{D_f^{enc}} + \underbrace{\kappa(1 - p_f)Pr(T_f > \Delta_f)}_{D_f^{cg}} + \underbrace{\kappa p_f}_{D_f^{loss}}. \quad (7)$$

For a given set of paths, the end-to-end packet loss probability p_f and the end-to-end packet overdue probability $Pr(T_f > \Delta_f)$ for a session $f \in \mathcal{F}$ is determined by the corresponding link parameters p_{ij} and b_{ij} , as well as the network topology, (i.e., paths sharing common links). In [9], we showed how to compute $Pr(T_f > \Delta_f)$ based on the Chernoff bound [3].

The set of inequalities in (2) provides the bounds for feasible video rate for each session, which are determined by the specific video sequence and the rate-distortion model. The inequalities in (3) represent the stability condition which ensures that the rate of the average aggregate traffic load on the link is less than the link capacity. (5) provides an expression to compute the parameter s_f^* . This expression is derived in [9] based on moment generating function for the end-to-end delay, and Chernoff bound.

The goal of the OPT-ARC problem is to obtain the best possible rates that would minimize (1) over a given set of paths for each video session. Minimizing (1) achieves the best utilization of network resources, while ensuring that the overall video quality is maximized for all the video sessions. Note that choosing a different objective function such as $\min \max \{D_f^e\}$ or an objective function in the form of a utility function $\sum_f f(D_f^e)$, does not change the solution procedure, which will be presented in the next section.

It is easy to observe that, the objective function (1) is a non-polynomial, non-convex function of continuous variables, $\{R_f\}_{f \in \mathcal{F}}$. Since non-polynomial problems are NP-hard in general, we conjecture that OPT-ARC problem in its current form, is likely to be NP-hard as well. In [9], we showed that this class of complex optimization problems can be solved using metaheuristic algorithms, such as Genetic Algorithms [1]. However, a metaheuristic approach cannot offer theoretical guarantees and in many cases, it is important to know the optimality gap of the solution obtained by the specific algorithm. In the following section, we present a branch-and-bound

OPT-ARC

Minimize

$$D = \sum_{f \in \mathcal{F}} \left\{ D_0 + \frac{\omega}{R_f - R_0} + \kappa \left\{ p_f + (1 - p_f) \left\{ \frac{e^{-s_f^* \Delta_f}}{s_f^* \delta_f(s_f^*) \sqrt{2\pi}} \right. \right. \right. \\ \left. \left. \left. \prod_{\{i,j\} \in L_f} \frac{b_{ij} - \sum_{\varphi \in \mathcal{F}} \left[\prod_{\{m,n\} \in \bar{L}_\varphi^{ij}} (1 - p_{mn}) \right] R_\varphi}{b_{ij} - \sum_{\varphi \in \mathcal{F}} \left[\prod_{\{m,n\} \in \bar{L}_\varphi^{ij}} (1 - p_{mn}) \right] R_\varphi - s_f^*} \right\} \right\} \right\} \quad (1)$$

subject to

$$\underline{R}_f \leq R_f \leq \bar{R}_f, \quad \forall f \in \mathcal{F} \quad (2)$$

$$\sum_{f \in \mathcal{F}} \left[\prod_{\{m,n\} \in \bar{L}_f^{ij}} (1 - p_{mn}) \right] \cdot R_f \leq (1 - \epsilon) \cdot b_{ij}, \quad \forall \{i, j\} \in L_f \quad (3)$$

$$\delta_f^2 = \sum_{\{i,j\} \in L_f} \frac{1}{(b_{ij} - \sum_{\varphi \in \mathcal{F}} \left[\prod_{\{m,n\} \in \bar{L}_\varphi^{ij}} (1 - p_{mn}) \right] R_\varphi - s_f^*)^2}, \quad \forall f \in \mathcal{F} \quad (4)$$

$$\Delta_f = \sum_{\{i,j\} \in L_f} \frac{1}{b_{ij} - \sum_{\varphi \in \mathcal{F}} \left[\prod_{\{m,n\} \in \bar{L}_\varphi^{ij}} (1 - p_{mn}) \right] R_\varphi - s_f^*}, \quad \forall f \in \mathcal{F}. \quad (5)$$

solution procedure for OPT-ARC problem. Our proposed solution procedure can produce a solution within an error of ϵ to the global optimum, where $0 < \epsilon < 1$ is an arbitrarily small value reflecting the tolerance in approximation.

III. SOLUTION PROCEDURE

In this section, we first discuss the branch-and-bound framework, and then, present details of embedding the RLT-relaxation of the original problem into this branch-and-bound framework.

A. Branch-and-Bound

Branch-and-bound is an iterative relaxation algorithm [10], that seeks to produce an ϵ -optimal solution to a non-linear programming (NLP) problem, by partitioning the original solution space into sub-hyperrectangles. 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) [11] to obtain a linear programming (LP) relaxation (see Section III-B). 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* is employed to obtain a feasible solution to the original NLP problem, using the infeasible lower bounding solution as a starting point. 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-B. 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 and upper bounds for P are computed as

$$\begin{cases} LB = \min\{LB_k : \text{Problem } k \in L\} \\ UB = \min\{UB_k : \text{Problem } k \in L\}. \end{cases} \quad (8)$$

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. At any given iteration, if $LB \geq (1 - \epsilon) \cdot UB$, the procedure exits with an ϵ -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 ϵ -optimal 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 *fathom* certain branches of the branch-and-bound tree, without solving all the nodes in the branch to completion. Depending on the effectiveness of this pruning strategy, the branch-and-bound procedure can provide a solution much faster than a general divide-and-conquer approach.

B. 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 [11]. 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. In our approach, the key to obtaining an ϵ -optimal solution is to embed RLT into the branch-and-bound framework as discussed earlier, and to coordinate with it a suitable partitioning strategy, so that the gap between the lower and upper bounds thus generated converges iteratively to zero.

It is worth noting that, when incorporating RLT into the branch-and-bound framework, the original bounding set Ω 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*. In our solution procedure, the branching variable is chosen to be the 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.

In the following section, we reformulate the OPT-ARC problem into a quadratic optimization problem. Then, we replace all the non-linear terms as discussed above and add the corresponding RLT constraints into the problem formulation. We obtain the *linear programming relaxation* problem (ℓ -ARC).

C. Reformulation

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

We can transform the first set of non-polynomial terms in the objective function (1) by defining new variables $u_f = 1/(R_f - R_0)$. Substituting u_f into the objective function, we obtain linear terms $\omega \cdot u_f$, and a set of new polynomial constraints $u_f \cdot (R_f - R_0) = 1$. We follow a similar procedure in order to transform the second non-polynomial term in the objective function. Recall that α_{ij} denotes the available bandwidth on a link $\{i, j\}$. Defining new substitution variables g_f and v_{ij} , where $v_{ij} = \frac{1}{\alpha_{ij} - s_f^*}$ and $g_f = \Pr(T_f > \Delta_f)$, we have,

$$g_f = \left[\frac{e^{-s_f^* \Delta_f}}{s_f^* \delta_f \sqrt{2\pi}} \prod_{\{i,j\} \in L_f} \alpha_{ij} \cdot v_{ij} \right]. \quad (9)$$

This product form motivates us to apply logarithms on both sides of (9), which will lead to a linear constraint. As a result, the complexity on the objective function can be effectively

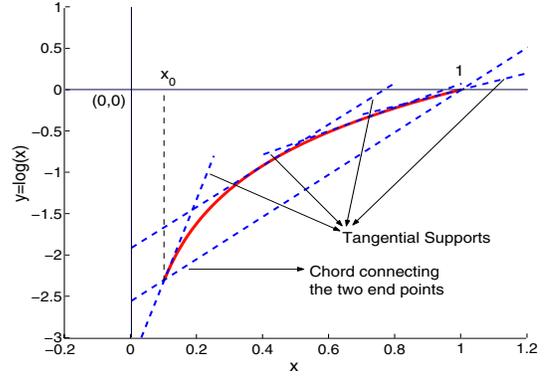


Fig. 1. Polyhedral outer approximation for $y = \log(x)$ in $0 < x_0 \leq x \leq 1$.

moved into the constraints. We use similar approach to convert all other non-polynomial terms in the constraints to a polynomial form of order two.

Once the objective function and constraints are simplified, we consider 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* in concert with several tangential supports. A four-point tangential approximation can be obtained as illustrated in Figure 1. The corresponding convex envelope consists of a chord connecting the two end points, which is used in combination with four tangential supports at four points including the two end points. As a result, every logarithmic relationship obtained from (9) translates to five linear constraints constituting a polyhedral outer approximation. Note that such polyhedral outer approximations will be iteratively tightened during the branch-and-bound procedure (see Section III-A).

Problem OPT-ARC is now transformed into a polynomial NLP problem of the second order. The objective function (1) is linearized and the complexity is shifted into the constraints in the quadratic form.

IV. NUMERICAL RESULTS

In this section, we present the simulation results for the optimal rate control problem for video transport. In each simulation setting, a wireless ad hoc network is generated by placing a number of nodes at random locations in a rectangular region. As discussed before, we assume that we are given a set of pre-selected paths, each path connecting a source-destination pair. 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 400-frame “Foreman” trace in the quarter common intermediate format (QCIF). The video was encoded with an intra rate of 1/15 and a frame rate of 30 fps. The rate-distortion parameters are obtained from [12]. 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 ℓ -ARC. At every node in the branch-and-

TABLE II
PERFORMANCE OF THE PROPOSED SOLUTION PROCEDURE.

Case	Initial Feasible Solution.	ϵ -optimal solution	ϵ
I	131.69	119.90	0.05
II	129.26	117.09	0.05
III	137.14	128.34	0.05

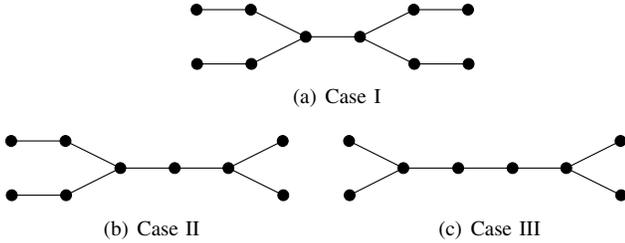


Fig. 2. Path topology.

bound tree, the local search algorithm discussed in Section III-A is used to obtain a feasible solution from the solution to the LP relaxation.

A. Performance on Different Instances of Problem OPT-ARC

We first examine the performance of the proposed solution procedure with different instances of Problem OPT-ARC, which are presented in Table II. In each case, the same 50-node network sustains two source destination pairs, and the number of shared links are increased from one to three as shown in Figure 2. The decoding deadline is 0.2 s for all the cases in Table II.

The second column of Table II presents the total distortion values found by solving the corresponding UB for the root node in the branch-and-bound tree and is termed as “Initial Feasible Solution”. The third column of Table II presents the ϵ -optimal solution found by the algorithm, and the last column shows the value of ϵ used in the termination of the branch-and-bound tree. As we can see, 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. Impact of Link Capacity

In the rest of this section, we present a study on the impact of link capacity on the performance of the proposed rate-allocation scheme. For this experiment, we use a 50-node network with two video sessions and the path topology as shown in Figure 2(a) i.e., the sessions share a common link. Figure 3 plots the optimal rates allocated to each session as computed by the algorithm, by varying the link capacity of the shared link. More specifically, we vary the link capacity of the common link from 100 Kb/s to 500 Kb/s on the x-axis, and then plot the corresponding rate allocated to each session on the y-axis. Figure 4 plots the average distortion found by the proposed algorithm for the same values of the

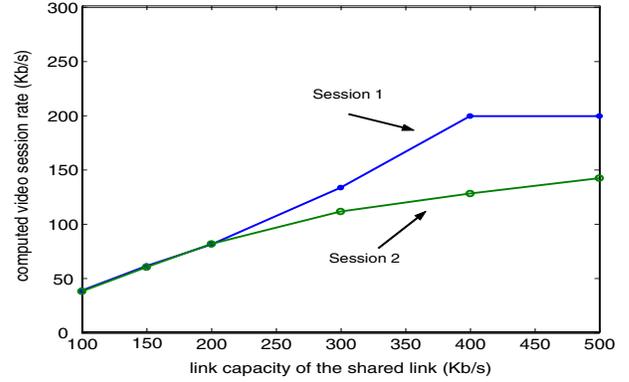


Fig. 3. Video rate for each session for varying shared link capacity.

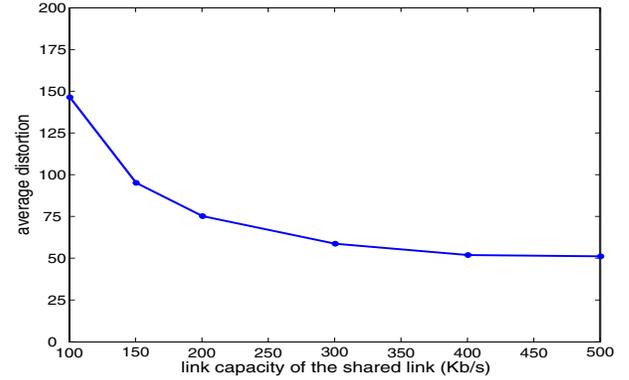


Fig. 4. Average distortion under varying shared link capacity.

shared link capacity as in Figure 3. Two observations can be made here. First, when the capacity of the shared link is very high (500 Kb/s), the video rates allocated to each session are far apart. However, as the shared link capacity is reduced, the rates come closer and finally for a link capacity of 100 Kb/s, the rates allocated to each session are almost identical. Second, the average distortion is low and relatively constant for high capacities of the shared link, and as the link capacity reduces, the average distortion increases rapidly.

These observations can be explained as follows. As long as at least one session is not bottlenecked by the shared link, the video rates for each session could be different. This is evident in Figure 2(a) when the link capacity of the bottleneck link is between 500 Kb/s and about 250 Kb/s. Here, the bottleneck link of one or both sessions could be different from the shared link and also, the bottleneck link of one session is higher than that of the other session. Also, between 400 Kb/s and 500 Kb/s, the bottleneck capacity of session 1 reaches its maximum value, and so it remains constant in this range. However, around 200 Kb/s, the optimal video rates clearly converge to a value of around 80 Kb/s for each session, indicating that the shared link is now the bottleneck link for both sessions.

In order to explain the second observation, recall that the end-to-end distortion D_f^e consists of three components: $D_f^e = D_f^{enc} + D_f^{cg} + D_f^{loss}$. Since the path loss probabilities remain

the same, D_f^{loss} does not change. Each of the remaining two terms will dominate for different ranges of the link capacity. For large link capacity, D_f^{enc} is the dominating component since the congestion in the network is negligible. Recall that D_f^{enc} is given by $D_0 + \frac{\omega}{R_f - R_0}$. Based on this expression, we can see that beyond a certain threshold, any increase in rate R_f reduces D_f^{enc} only marginally. Therefore, in the case of large link capacity, the algorithm intelligently computes ϵ -optimal rates for each video session that fall beyond this threshold, so as to keep the average distortion minimum and relatively constant. For smaller values of link capacity, D_f^{cg} takes over, as the network suffers from severe congestion, increasing the the average distortion rapidly.

V. RELATED WORK

The idea of rate control for streaming video applications has been studied extensively in both wired and wireless domains [7], [14]. These efforts predominantly associate rate control with the transport layer. A lot of work has been done on applying equation based TCP-friendly rate control to video streaming in the wireless domain. Prior research (e.g., [2]) also showed the necessity of differentiating packet loss with congestion in wireless networks. However, they are usually based on optimizing network layer parameters, and do not explicitly consider any application layer behavior (refer [7] for a survey on video delivery over wireless networks). Cross-layer design in wireless networks has also received much attention in the context of elastic data as well as video applications [5], [6], [8], [13]. Typically, such designs consider jointly optimizing two or more of the following: power control, MAC, routing, scheduling and source coding, in order to make use of the network resources efficiently. Theoretical optimization formulations tend to benefit from some of the commonly used assumptions that the objective function is strictly concave, non decreasing and continuously differentiable [8]. However, in this paper, we show that when modeling the user perceived video quality as a function of network parameters like delay, congestion, etc., the objective function is considerably more complex and does not follow the assumptions discussed earlier. It is also worth noting that this class of prior efforts focus on a single video session while our work considers the interaction among multiple sessions.

VI. CONCLUSIONS

In this paper, we studied the problem of optimal rate control for video transport in multihop wireless networks. We presented an optimization problem that captured the tradeoff between the encoded video rate and the final quality of the reconstructed video at the receiver in terms of network layer parameters, in the presence of competing video sessions. The objective function minimized the collective distortion of all video sessions, in a solution space of bounded rates for each session. The solution procedure based on branch-and-bound framework and RLT was shown to be effective in generating ϵ -optimal solutions for the problem. We believe that this

solution procedure can be tuned effectively in order to study the performance limits for this class of optimization problems.

ACKNOWLEDGMENTS

We would like to thank Ramana Rao Kompella for his helpful comments and suggestions. Sastry Kompella is supported by an IGERT grant from the National Science Foundation (award DGE-9987586).

REFERENCES

- [1] T. Back, D. Fogel, and Z. Michalewicz, editors. *Handbook of Evolutionary Computation*. Oxford University Press, New York, NY, 1997.
- [2] S. Cen, P.C. Cosman, and G.M. Voelker. End-to-end differentiation of congestion and wireless losses. *IEEE/ACM Transactions on Networking*, 11(5):703 – 717, Oct. 2003.
- [3] H. Chernoff. A measure of asymptotic efficiency for tests of a hypothesis based on the sum of observations. *Annals Math. Statist.*, 23:493–507, 1952.
- [4] T. Clausen and P. Jacquet. Optimized Link State Routing Protocol (OLSR). RFC 3626 (Experimental), October 2003.
- [5] R.L. Cruz and A.V. Santhanam. Optimal routing, link scheduling and power control in multihop wireless networks. In *Proc. IEEE INFOCOM*, pages 702–711, Mar. 2003.
- [6] T. ElBatt and A. Ephremides. Joint scheduling and power control for wireless ad hoc networks. *IEEE Trans. Wireless Comm.*, 3(1):74 – 85, Jan. 2004.
- [7] M. Etoh and T. Yoshimura. Advances in wireless video delivery. *Proc. of IEEE*, 93(1):111 – 122, Jan. 2005.
- [8] X. Lin and N.B. Shroff. Utility maximization for communication networks with multi-path routing. Technical report, Purdue University, 2004.
- [9] S. Mao, S. Kompella, Y.T. Hou, and H.D. Sherali. Routing for multiple concurrent video sessions in wireless ad hoc networks. In *Proc. IEEE ICC*, pages 1229–1235, Seoul, Korea, May 2005.
- [10] G.L. Nemhauser and L.A. Wolsey. *Integer and Combinatorial Optimization*. John Wiley & Sons, New York, NY, 1999.
- [11] H.D. Sherali and W.P. Adams. *A Reformulation-Linearization Technique for Solving Discrete and Continuous Nonconvex Problems*. Kluwer Academic Publisher, Boston, MA, 1999.
- [12] K. Stulmuller, N. Farberand, M. Link, and B. Girod. Analysis of video transmission over lossy channels. *IEEE Journal on Selected Areas in Communications*, 18(6):1012–1032, June 2000.
- [13] M. van der Schaar and M. Tekalp. Integrated multi-objective cross-layer optimization for wireless multimedia transmission. In *Proc. IEEE ISCAS*, pages 3543–3546, May 2005.
- [14] Q. Zhang, W. Zhu, and Y.-Q. Zhang. Network-adaptive rate control and unequal loss protection with TCP-friendly protocol for scalable video over internet. *J. VLSI Signal Proc. Syst.*, 34(1-2):67–81, 2003.