On Routing for Multiple Description Video Over Wireless Ad Hoc Networks

Shiwen Mao, *Member, IEEE*, Y. Thomas Hou, *Senior Member, IEEE*, Xiaolin Cheng, *Student Member, IEEE*, Hanif D. Sherali, Scott F. Midkiff, *Senior Member, IEEE*, and Ya-Qin Zhang, *Fellow, IEEE*

***Abstract*—We study the problem of multipath routing for double description (DD) video in wireless ad hoc networks. We follow an *application-centric* cross-layer approach and formulate an optimal routing problem that minimizes the application layer video distortion. We show that the optimization problem has a highly complex objective function and an exact analytic solution is not obtainable. However, we find that a meta-heuristic approach such as *genetic algorithms* (GAs) is eminently effective in addressing this type of complex cross-layer optimization problems. We provide a detailed solution procedure for the GA-based approach. Simulation results demonstrate the superior performance of the GA-based approach versus several other approaches. Our efforts in this work provide an important methodology for addressing complex cross-layer optimization problems, particularly those involved in the application and network layers.**

***Index Terms*—Ad hoc networks, cross-layer design, metaheuristics, multipath routing, multiple description video, optimization.**

I. Introduction

WIRELESS ad hoc networks are characterized by the absence of traditional infrastructural support (e.g., base stations). As a result, information exchange among mobile nodes is achieved through multi-hop wireless communications. As progress in wireless ad hoc networking continues, there is an increasing expectation on enabling content-rich multimedia communications in such networks, due to the fact that real-time multimedia (e.g., live video) is far more substantive than simple data communications. However, at present, there are significant technical barriers that hinder the widespread deployment of multimedia applications in wireless ad hoc networks. In fact, what makes traditional single stream coding and layered coding successful in the Internet and certain wireless networks is the existence of a relatively stable path during the video session. Consequently, packet loss on important information (e.g., base layer) is kept low, and can be effectively handled by error control and concealment mechanisms. This is important since, for layered video, the successful reconstruction of video relies on the base layer, and the decoding of enhancement layers hinges upon lower enhancement layers as well as upon the base layer. However, this situation hardly holds true in wireless ad hoc networks, where there may not exist any single reliable path and packet loss may be beyond the recovery capability of most error control mechanisms.

Recently, *multiple description* (MD) coding has become a popular coding technique for media streaming [1]. With MD coding, multiple *equivalent* streams (or descriptions) are generated for a video source, such that *any* received subset of these streams can be used to reconstruct the original video, yielding a quality commensurate with the number of received descriptions. It has been recognized that MD coding matches perfectly with the wireless ad hoc network environment for multimedia applications [2]. This is because the topology of such networks is intrinsically mesh, within which multiple paths exist between any source and destination pair. Although the paths in such networks are fragile (i.e., will not remain reliable for an extended period of time), as long as the link/node failure events on different paths are not entirely correlated, the probability of concurrent loss of all of the descriptions will be low. Therefore, MD coding will remain effective, with improved video quality as more descriptions are received.

Several researchers have proposed to use MD coding with *multipath routing* for multimedia transport [2]–[5]. These interesting works have successfully demonstrated the efficacy of using MD with multipath routing, assuming that the set of paths is given *a priori*. However, the difficult problem of finding best paths for the descriptions has not been adequately addressed. In a recent work [6], Begen *et al.* studied the problem of multipath selection for double description (DD) video in the context of Internet overlay networks. The path selection problem is, however, solved via an exhaustive search.

In this paper, we study the problem of multipath routing for DD video in wireless ad hoc networks. We follow a *cross-layer* approach in problem formulation by considering the application layer performance (i.e., average video distortion) as a function of network layer performance metrics (e.g., bandwidth, loss, and path correlation). We show that the objective function is a complex ratio of high-order exponentials which is nondecomposable. Consequently, it would be futile to develop a tractable analytic solution. However, we find that a metaheuristic technique such as *genetic algorithms* (GAs) [7] is eminently suitable in addressing such type of complex cross-layer optimization problems. This is because GAs possess an intrinsic capability of handling a *population* of solutions (rather than working

Manuscript received May 6, 2005; revised November 21, 2005. This work was supported in part by the National Science Foundation under Grants ANI-0312655, CNS-0347390, and DMI-0094462, and by the Office of Naval Research under Grants N00014-03-1-0521 and N00014-05-1-0179. Part of this work was presented at IEEE INFOCOM 2005, March 2005, Miami, FL. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Yong Rui.

S. Mao, Y. T. Hou, X. Cheng, and S. F. Midkiff are with the Department of Electrical and Computer Engineering, Virginia Polytechnic Institute and State University, Blacksburg, VA 24061 USA (e-mail: smao@ieee.org; thou@vt.edu; xicheng@vt.edu; midkiff@vt.edu).

H. D. Sherali is with the Department of Industrial and Systems Engineering, Virginia Polytechnic Institute and State University, Blacksburg, VA 24061 USA (e-mail: hanifs@vt.edu).

Y.-Q. Zhang is with Microsoft Corporation, Redmond, WA 98502 USA (e-mail: yzhang@microsoft.com).

Digital Object Identifier 10.1109/TMM.2006.879845

with a single current solution during each iteration). Such capability gives GAs the unique strength of identifying promising regions in the search space (not necessarily convex) and having less of a tendency to be trapped in a local optimum, as compared with other trajectory-based metaheuristics (e.g., *simulated annealing* (SA) and *tabu search* (TS) [8]). Using numerical results, we show that significant performance gains can be achieved by the GA-based approach over trajectory-based approaches. In order to examine the quality of GA solutions, as well as setting its termination conditions, we develop a simple but tight lower bound on video distortion, which has similar computational complexity as Dijkstra's algorithm. Finally, we show that the GA-based multipath routing can be incorporated into many existing distributed ad hoc network routing protocols (e.g., [9], [10]), particularly the class of *proactive* protocols. As an example, we present a distributed implementation based on the Optimized Link State Routing Protocol (OLSR) [10].

The remainder of this paper is organized as follows. In Section II, we formulate a cross-layer optimization problem for DD video over multiple paths in ad hoc networks. Section III presents a lower bound for video distribution. In Section IV, we describe our GA-based approach and numerical results are presented in Section V. Section VI describes a distributed implementation of the proposed approach. Section VII discusses related work, and Section VII concludes this paper.

TABLE I
NOTATION

Symbols	Definitions
$\mathcal{G}\{V,E\}$:	graph representation of the network
V :	set of vertices in the network
E :	set of edges in the network
s :	source node
t :	destination node
$\mathcal{P}$:	a path from s to t
g_i :	an intermediate node in a path
$\{i,j\}$:	a link from node i to node j
b_{ij} :	bandwidth of link $\{i,j\}$
p_{ij} :	success probability of link $\{i,j\}$
l_{ij} :	average length of loss burst on link $\{i,j\}$
R_h :	rate of description h in bits/sample
d_0 :	distortion when both descriptions are received
d_h :	distortion when only Description h is received $h=1,2$
D :	average distortion
T_{on} :	average "up" period of the joint links
P_{00} :	probability of receiving both descriptions
P_{01} :	probability of receiving description 1 only
P_{10} :	probability of receiving description 2 only
P_{11} :	probability of losing both descriptions
x_{ij}^h :	routing index variables, defined in (9)
α_{ij}:	"up" to "down" transition prob. of link $\{i,j\}$
β_{ij} :	"down" to "up" transition prob. of link $\{i,j\}$
p_{jnt} :	average success prob. of joint links
p_{dj}^h :	average success prob. of disjoint links on $\mathcal{P}_h$
B_{jnt} :	minimum bandwidth of the shared links
θ :	GA crossover rate
μ :	GA mutation rate

II. Problem Description

An ad hoc network can be modeled as a stochastic directed graph $\mathcal{G}\{V,E\}$, where V is the set of vertices and E the set of edges. We assume that nodes are reliable during the video session, but links may fail with certain probabilities. Accurate and computationally efficient characterization of an end-to-end path in a wireless ad hoc network with consideration of mobility, interference, and the time-varying wireless channels is extremely difficult and remains an open problem. As an initial step, we focus on the network layer characteristics in this paper, assuming that the physical and MAC layer dynamics of wireless links are translated into network layer parameters. For example, we could characterize a link $\{i,j\} \in E$ by

- b_{ij}: the available bandwidth of link $\{i,j\}$. We assume that the impact of other traffic sessions are accounted for through the link available bandwidth;
- p_{ij}: the probability that link $\{i,j\}$ is "up";
- l_{ij}: average burst length for packet losses on link $\{i,j\}$.

In practice, these parameters can be measured by every node, and distributed throughout the network using Link State Advertisements (LSA) [10]. We focus on the bandwidth and failure probabilities of a path, since these two are key characteristics for data transmission, as well as the most important factors that determine video distortion [see (2)]. Other link characteristics, such as delay, jitter, congestion, and signal strength could be incorporated into this framework as well (e.g., see [11]). Table I lists the notation used in this paper.

A. Rate-Distortion Regions for DD Coding

Throughout this paper, we use double-description coding for MD video. We consider double-description video since it is most widely used in practice [2]–[6]. In general, using more descriptions and paths will increase the robustness to packet losses and path failures. However, more descriptions may increase the video bit rate for the same video quality. The study in [12] demonstrates that the most significant performance gain is achieved when the number of descriptions increases from one to two, with only marginal improvements achieved for further increases in number of descriptions.

For video coding and communications, a distortion rate model describes the relationship between the achieved distortion and the bit rate (i.e., the quality and the length of the representation). For two descriptions (each generated for a sequence of video frames), 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. Let R_h denote the rate in bits/pixel of Description h, $h = 1, 2$. The rate-distortion region for a memoryless *i.i.d.* Gaussian source with the square error distortion measure was first introduced in [13]. For computational efficiency, Alasti *et al.* in [14] introduce the following distortion-rate function for studying the impact of congestion on DD coding (also used in this paper).

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

where σ^2 is the variance of the source. Let P_{00} be the probability of receiving both descriptions, P_{01} the probability of receiving Description 1 only, P_{10} the probability of receiving Description 2 only, and P_{11} the probability of losing both descriptions. Then, the average distortion of the received video can be approximated as

$$D = P_{00} \cdot d_0 + P_{01} \cdot d_1 + P_{10} \cdot d_2 + P_{11} \cdot \sigma^2. \tag{2}$$

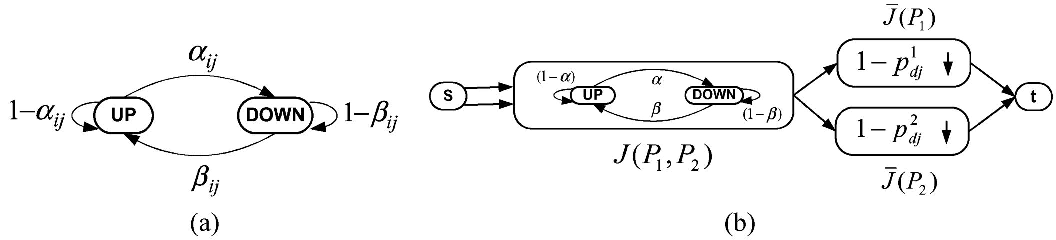


Fig. 1. Link and path models. (a) The Gilbert two-state link model; (b) a simplified path model for double-description video.

Finding the rate distortion region for MD video is still an open research problem [1]. The MD region is well understood only for memoryless Gaussian sources with squared-error distortion measure, which bounds the MD region for any continuous-valued memoryless source with the same distortion measure. Although there are several empirical models used in the literature, these models are dependent on the specific video sequence, and the model parameters are determined by using regression techniques [4], [6]. Therefore, these models are not entirely suitable for our cross-layer routing algorithm, which is not specific to a particular stored video or MD coding technique. More importantly, we believe that such a model should be robust and effective for live video. Our simulation results show that although the distortion-rate function in (1) is an approximation for DD video, significant improvement in received video quality could be achieved over alternative approaches by incorporating (2) into the optimal routing problem formulation (see Section V). It is also worth noting that our formulation does not depend on any specific distortion-rate function. A more accurate distortion-rate function for MD video could be easily incorporated into our formulation should it be available in the future.

B. Description Rates and Success Probabilities

As a first step to formulate the problem of optimal multipath routing, we need to know how to compute the average distortion D as a function of link statistics for a *given* pair of paths. That is, we need to compute the end-to-end bandwidth (or rate) for each stream and joint probabilities of receiving the descriptions (see (1) and (2)).

For a source-destination pair $\{s,t\}$, consider two given paths $[\mathcal{P}_1,\mathcal{P}_2]$ in $\mathcal{G}\{V,E\}$. Since we do not mandate "disjointedness" in routing, $\mathcal{P}_1$ and $\mathcal{P}_2$ may share nodes and links. Similar to the approach in [4] and [6], we classify the links along the two paths 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 of disjoint links on the two paths, denoted as $\bar{\mathcal{J}}(\mathcal{P}_h)$, $h=1,2$, respectively. Then, the minimum bandwidth of $\mathcal{J}(\mathcal{P}_1,\mathcal{P}_2)$, B_{jnt}, is

$$B_{jnt}=\begin{cases}\min_{\{i,j\}\in\mathcal{J}(\mathcal{P}_1,\mathcal{P}_2)}\{b_{ij}\}, & \text{if } \mathcal{J}(\mathcal{P}_1,\mathcal{P}_2)\neq\emptyset\\ \infty, & \text{otherwise.}\end{cases}$$

The rates of the two video streams, R_1 and R_2, can be computed as

$$\begin{cases}R_h=\rho\cdot B(\mathcal{P}_h), & \text{if } \sum_{m=1}^{2}B(\mathcal{P}_m)\leq B_{jnt}, h=1,2\\ R_1+R_2\leq\rho\cdot B_{jnt}, & \text{otherwise}\end{cases}\tag{3}$$

where $B(\mathcal{P}_h)=\min_{\{i,j\}\in\mathcal{P}_h}\{b_{ij}\}, h=1,2,$ and ρ is a constant determined by the video format and frame rate. 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. For the quarter common intermediate format (QCIF) [176 $\times$ 144 Y pixels/frame, 88 $\times$ 72 Cb/Cr pixels/frame], we have $\kappa=1.5$ and $\rho=1/(1.5\cdot 176\cdot 144\cdot f)$. The first line in (3) is for the case when the joint links are not the bottleneck of the paths. The second line of (3) is for the case where one of the joint links is the bottleneck of both paths. In the latter case, we assign the bandwidth to the paths by splitting the bandwidth of the shared bottleneck link in proportion to the mean success probabilities of the two paths, while an alternative approach is to split the bandwidth evenly for balanced descriptions.

We now focus on how to compute the end-to-end success probabilities. 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 in the disjoint portions. Therefore, the success probabilities on the disjoint portions of the two paths are

$$p_{dj}^h=\begin{cases}\prod_{\{i,j\}\in\bar{\mathcal{J}}(\mathcal{P}_h)}p_{ij}, & \text{if } \bar{\mathcal{J}}(\mathcal{P}_h)\neq\emptyset, h=1,2\\ 1, & \text{otherwise, } h=1,2.\end{cases}\tag{4}$$

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 Fig. 1(a). With this model, there is no packet loss when the link is in the "up" state; all packets are dropped when the link is in the "down" state. 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 follow the well-known Fritchman model [15] in modeling the aggregate process as an on-off process. Since a packet is successfully delivered on the joint portion if and only if all joint links are in the "up" state, we can 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

$$T_{on}=\frac{1}{1-\prod_{\{i,j\}\in\mathcal{J}(\mathcal{P}_1,\mathcal{P}_2)}(1-\alpha_{ij})}.\tag{5}$$

If the joint link set is not empty, the probability of a successful delivery on the joint links can be written as

$$p_{jnt} = \begin{cases} \prod_{\{i,j\} \in \mathcal{J}(\mathcal{P}_1,\mathcal{P}_2)} p_{ij}, & \text{if } \mathcal{J}(\mathcal{P}_1,\mathcal{P}_2) \neq \emptyset \\ 1, & \text{otherwise.} \end{cases} \tag{6}$$

Finally, the transition probabilities of the aggregate on–off process are

$$\alpha = \frac{1}{T_{on}} \text{ and } \beta = \frac{p_{jnt}}{T_{on}(1-p_{jnt})}. \tag{7}$$

Note that $\alpha = 0$ and $\beta = 0$ if $\mathcal{J}(\mathcal{P}_1,\mathcal{P}_2) = \emptyset$. The consolidated path model is illustrated in Fig. 1(b), where $\mathcal{J}(\mathcal{P}_1,\mathcal{P}_2)$ is modeled as a two-state Markov process with parameters $\{\alpha,\beta\}$, 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

$$\begin{cases} P_{00} = p_{jnt} \cdot (1-\alpha) \cdot p_{dj}^1 \cdot p_{dj}^2 \\ P_{01} = p_{jnt} \cdot p_{dj}^1 \cdot \left[1-(1-\alpha)\cdot p_{dj}^2\right] \\ P_{10} = p_{jnt} \cdot \left[1-(1-\alpha)p_{dj}^1\right] \cdot p_{dj}^2 \\ P_{11} = 1 - p_{jnt} \cdot \left[p_{dj}^1 + p_{dj}^2 - (1-\alpha)\cdot p_{dj}^1 \cdot p_{dj}^2\right]. \end{cases} \tag{8}$$

C. The Optimal Multipath Routing Problem

With the above preliminaries, we now set out to formulate the multipath routing problem for MD video. To characterize any $s-t$ path $\mathcal{P}_h$, we define the following binary variables:

$$x_{ij}^h = \begin{cases} 1, & \text{if } \{i,j\} \in \mathcal{P}_h \\ 0, & \text{otherwise.} \end{cases} \tag{9}$$

With these variables, an arbitrary path $\mathcal{P}_h$ can be represented by a vector $\mathbf{x}^h$ of $|E|$ elements, each of which corresponds to a link and has a binary value. We can formulate the problem of multipath routing for MD video (OPT-MM) as follows.

OPT-MM

$$\textbf{Minimize}: \ D = P_{00}\cdot d_0 + P_{01}\cdot d_1 + P_{10}\cdot d_2 + P_{11}\cdot\sigma^2 \tag{10}$$

subject to:

$$\sum_{j:\{i,j\}\in E} x_{ij}^h - \sum_{j:\{j,i\}\in E} x_{ji}^h = \begin{cases} 1, & \text{if } i = s, & i \in V, h = 1,2 \\ -1, & \text{if } i = t, & i \in V, h = 1,2 \\ 0, & \text{otherwise}, & i \in V, h = 1,2 \end{cases} \tag{11}$$

$$\sum_{j:\{i,j\}\in E} x_{ij}^h \begin{cases} \leq 1, & \text{if } i \neq t, & i \in V, h = 1,2 \\ = 0, & \text{if } i = t, & i \in V, h = 1,2 \end{cases} \tag{12}$$

$$x_{ij}^1 \cdot R_1 + x_{ij}^2 \cdot R_2 \leq \rho \cdot b_{ij}, \quad \{i,j\} \in E \tag{13}$$

$$x_{ij}^h \in \{0,1\}, \quad \{i,j\} \in E, \ h = 1,2. \tag{14}$$

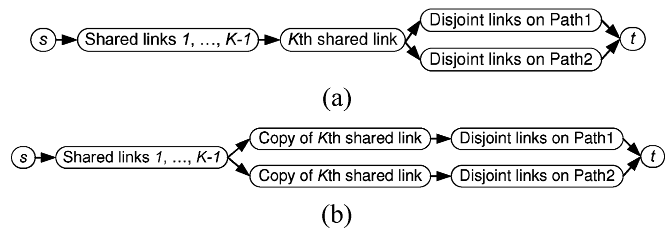


Fig. 2. The two solutions have the same set of links. The only difference between them is that a link is shared in $\hat{x}$ (the K-th shared link), but not shared in $\bar{x}$ (a copy of which is appended to each of the disjoint portions). (a) Solution $\hat{x}$; (b) solution $\bar{x}$.

In Problem OPT-MM, $\{x_{ij}^h\}$ are binary optimization variables. Constraints (11) and (12) guarantee that the paths are loop-free, while constraint (13) guarantees 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), (3), and (8). Different statistics of a given pair of path have different impact on the received video distortion. Specifically, for larger end-to-end bandwidth, the video rate is be higher and there is less distortion due to the lossy coder (i.e., d_0, d_1, and d_2 are all decreasing functions of the description rates). With a lower end-to-end loss rate, fewer video frames will be corrupted. This is modeled in (10), where σ^2 is usually much larger than d_0, d_1, and d_2, and d_h is usually larger than d_0, $h = 1{,}2$. Finally, the impact of path correlation is actually considered in the derivation of the joint probabilities of receiving the description. In Problem OPT-MM, all the three elements are integrated in the objective function (10), and are jointly optimized in routing.

The objective function (10) is a highly complex ratio of high-order exponentials of the x-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 [or can be conditioned on the exceedingly complex products of the binary factors x_{ij}^1 and $(1-x_{ij}^1)$ with x_{ij}^2 and $(1-x_{ij}^2)$]. In [16], Sherali *et al.* considered a problem that seeks a pair of disjoint paths in a network such that the total travel time over the paths is minimized, where the travel time on a link might be either a constant, or a nondecreasing (or unstructured) function of the time spent on the previous links traversed. Even for a simple special case where all the links except one have a constant travel time (and hence linear objective terms), this problem is shown to be NP-hard. Our problem has much more complex relationships pertaining to the contribution of each individual link to the objective function, which depends in general on the other links that are included in a fashion that has no particular structural property such as convexity. Hence, it is likely to be NP-hard as well. However, we leave a rigorous proof of this NP-hardness to a separate paper.

III. A Lower Bound for Distortion

Before describing our GA-based approach, we first construct a lower bound on the achievable video distortion. Such a bound will be useful in evaluating the performance of a heuristic algorithm, as well as serving as a reference for setting termination conditions for iterative algorithms.

ALG-LB:
0. Find the maximum end-to-end bandwidth, b^*, among all s–t paths;
1. Find the maximum end-to-end success probability, p^*, among all s–t paths;
2. Construct a solution $x_l^* = \{\mathcal{P}_1^l, \mathcal{P}_2^l\}$, satisfying:
3. (1) $\mathcal{P}_1^l$ is disjoint with $\mathcal{P}_2^l$;
4. (2) $B(\mathcal{P}_h^l) = b^*$, $h = 1, 2$;
5. (3) $p_{dj}^h(\mathcal{P}_h^l) = p^*$, $h = 1, 2$.

Fig. 3. A procedure to construct a lower bounding solution x_l^*.

We find that the average video distortion D possesses the following *monotonicity* properties.

M1: D is nonincreasing with R_h, $h = 1,2$.

M2: For two completely disjoint paths, D is nonincreasing with p_{dj}^h, $h = 1,2$.

M3: Consider the two solutions $\hat{x}$ and $\bar{x}$ shown in Fig. 2. Assume i) the two solutions provide the same description rates (i.e., the Kth shared link is not the bottleneck link of the two paths); and ii) the on–off failure process of the Kth shared link is random or bursty, i.e., $\alpha_K + \beta_K \leq 1$. Then $D(\hat{x}) \geq D(\bar{x})$.

Properties M1 and M2 are derived from the fact that the first derivatives of D with regard to the description rates and p_{dj}^h are all less than or equal to zero. The assumption in Property M3 relates to the covariance of two consecutive failure events, denoted as X_k and X_{k+1}, on link $\{i, j\}$:

$$\mathrm{Cov}\{X_k, X_{k+1}\} = \frac{\alpha_{ij}\beta_{ij}}{(\alpha_{ij} + \beta_{ij})^2}(1 - \alpha_{ij} - \beta_{ij}). \qquad (15)$$

If $\alpha_{ij} + \beta_{ij} < 1$, two successive failures (or losing both descriptions sent back to back on this link) are positively correlated, i.e., the failure process is *bursty*, which, we argue, is not atypical in wireless ad hoc networks. When $\alpha_{ij} + \beta_{ij} = 1$, two successive failures are un-correlated, corresponding to *random* packet losses. When $\alpha_{ij} + \beta_{ij} > 1$, the successive failures are negatively correlated (called *sub-bursty*), which should be rare in wireless ad hoc networks. In Fig. 2, if the Kth shared link has bursty or random losses, then $\bar{x}$ yields a distortion no higher than $\hat{x}$. The proof for Property M3 is presented in Appendix I.

We are now ready to construct a simple but tight lower bound on the average video distortion. Algorithm ALG-LB in Fig. 3 is such an algorithm. In Fig. 3, ALG-LB first determines the optimal end-to-end bandwidth b^* and the optimal end-to-end success probability p^*. It then constructs two *virtual* paths, which yield a distortion lower bound. Since we are interested in a lower bound, the corresponding physical paths are not necessarily feasible. In ALG-LB, b^* can be found using, e.g., the algorithm in [17] with time complexity $O(|E| \cdot \log^* |V|)$, where $\log^* n$ is the *iterated logarithm function*; p^* can be found by setting link costs to $\log(1/p_{ij})$, $\forall\{i, j\} \in E$, and then applying Dijkstra's algorithm to find the path having the minimum cost. The time complexity of finding p^* is $O(|E| + |V| \cdot \log |V|)$.

Proposition 1: The distortion $D(x_l^*)$ constructed by ALG-LB is a lower bound for distortion D defined in (10).

A proof of Proposition 1 is given in Appendix I-B. Although we show that x_l^* dominates all disjoint and joint feasible solutions, it does not necessarily imply that the optimal paths are *always* disjoint. The two optimal paths may share a "good" link in order to avoid the use of low quality links. Furthermore, $D(x_l^*)$ becomes an *exact* bound, i.e., $D(x_l^*) = D(x^*)$, if x_l^* is realizable. We will illustrate the tightness of this lower bound in Section V.

IV. A Metaheuristic Approach

Although the lower bound offered by ALG-LB provides a good estimate for D, it does not yield a pair of feasible paths. In this section, we present a solution procedure that produces a pair of feasible and near-optimal paths.

We find that GAs [7] are eminently suitable for addressing this type of complex combinatorial problems, most of which are multimodal and nonconvex. GAs are *population-based* metaheuristic inspired by the *survival-of-the-fittest* principle. It has the intrinsic strength of dealing with a set of solutions (i.e., a population) at each step, rather than working with a single, current solution. At each iteration, a number of genetic operators are applied to the individuals of the current population in order to generate individuals for the next generation. In particular, GA uses genetic operators known as *crossover* to recombine two or more individuals to produce new individuals, and *mutation* to achieve a randomized self-adaptation of individuals. The driving force in GA is the *selection* of individuals based on their fitness (in the form of an objective function) for the next generation. The survival-of-the-fittest principle ensures that the overall quality of the population improves as the algorithm progresses from one generation to the next.

Fig. 4 displays the flow chart for our GA-based approach to the MD multipath routing problem. In what follows, we use an example ad hoc network shown in Fig. 5(a) to illustrate the components in our GA-based approach. The termination condition in Fig. 4 could be based on the total number of iterations (generations), maximum computing time, a threshold of desired video distortion, or a threshold based on the lower bound described in Section III.

1) Solution Representation and Initialization: In GAs, a feasible solution is encoded in the genetic format. For a routing problem, a natural encoding scheme would be to define a node as a gene. Then, an end-to-end path, consisting of an ordered sequence of nodes (connected by the corresponding wireless links), can be represented as a chromosome [18]. For Problem OPT-MM, each feasible solution consists of a pair of paths (i.e.,

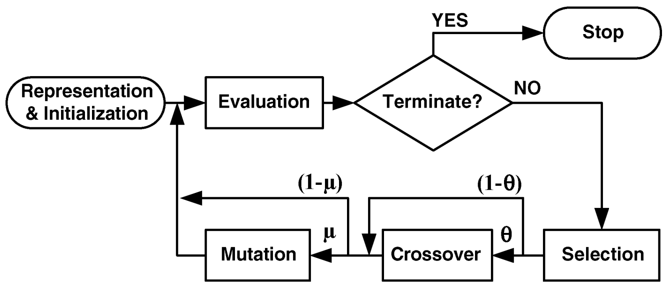
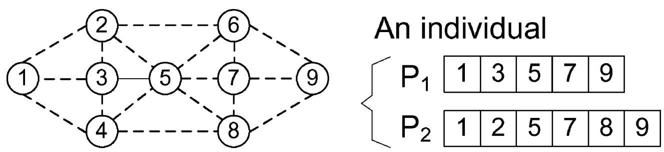


Fig. 4. Flow chart of the GA-based approach.



(a) (b)

Fig. 5. Example network and coding of an individual ($s = 1$ and $t = 9$). (a) Example ad hoc network; (b) example individual.

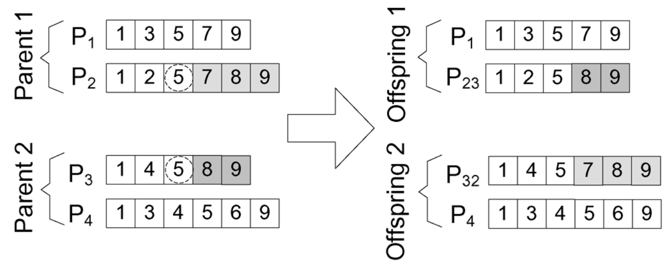


Fig. 6. Example of the crossover operation.

a pair of chromosomes), denoted as $[\mathcal{P}_1, \mathcal{P}_2]$. An individual in this case is a pair of vectors containing the nodes on paths $\mathcal{P}_1$ and $\mathcal{P}_2$ [see, e.g., Fig. 5(b)].

Before entering the main loop in Fig. 4, we need to generate an initial population, i.e., a set of solutions. A simple approach would be to generate this set of solutions by randomly appending feasible elements (i.e., nodes with connectivity) to a partial solution. Under this approach, each construction process starts with source node s. Then, the process randomly chooses a link incident to the current end-node of the partial path and appends the link with its corresponding head-node to augment the path, until destination node t is reached. It is important to ensure that the intermediate partial path is loop-free during the process. After generating a certain set of paths for $s-t$ independently, a population of individuals can be constructed by pairing paths from this set. Our numerical results show that a properly-designed GA is not very sensitive to the quality of the individuals in the initial population.

2) Evaluation: The fitness function $f(\bar{x})$ of an individual, $\bar{x} = [\mathcal{P}_1, \mathcal{P}_2]$, is closely tied to the objective function (i.e., distortion D). Since the objective is to minimize the average distortion function D, we have adopted a fitness function defined as the inverse of the distortion value, i.e., $f(\bar{x}) = 1/D(\bar{x})$. This simple fitness definition appears to work very well, although we intend to explore other fitness definitions in our future effort.

3) Selection: During this operation, GA selects individuals that have a better chance or potential to produce "good" offspring in terms of their fitness values. By virtue of the selection operation, "good" genes among the population are more likely to be passed to the future generations. We use the so called *Tournament* selection [7] scheme, which randomly chooses m individuals from the population each time, and then selects the best of these m individuals in terms of their fitness values. By repeating either procedure multiple times, a new population can be selected.

4) Crossover: Crossover mimics the genetic mechanism of reproduction in the natural world, in which genes from parents are recombined and passed to offspring. The decision of whether or not to perform a crossover operation is determined by the *crossover rate* θ.

Fig. 6 illustrates one possible crossover implementation. Suppose that we have two parent individuals $x_1 = [\mathcal{P}_1, \mathcal{P}_2]$ and $x_2 = [\mathcal{P}_3, \mathcal{P}_4]$. We could randomly pick one path in x_1 and one in x_2, say $\mathcal{P}_2$ and $\mathcal{P}_3$. If one or more common nodes exist in these two chosen paths, we could select the first such common node that exists in $\mathcal{P}_2$, say g_r, where $g_r \notin \{s, t\}$, and we can then concatenate nodes $\{s, \cdots, g_r\}$ from $\mathcal{P}_2$ with nodes $\{g_{r+1}, \cdots, t\}$ in $\mathcal{P}_3$ (where g_{r+1} denotes the next downstream node of g_r in $\mathcal{P}_3$) to produce a new path $\mathcal{P}_{23}$. Likewise, using the first such node $g_{r'}$ in $\mathcal{P}_3$ that repeats in $\mathcal{P}_2$ (which may be different from g_r), we can concatenate the nodes $\{s, \cdots, g_{r'}\}$ from $\mathcal{P}_3$ with the nodes $\{g_{r'+1}, \cdots, t\}$ in $\mathcal{P}_2$ to produce a new path $\mathcal{P}_{32}$. It is important that we check the new paths to be sure that they are loop-free. The two offspring generated in this manner are $[\mathcal{P}_1, \mathcal{P}_{23}]$ and $[\mathcal{P}_{32}, \mathcal{P}_4]$. On the other hand, if $\mathcal{P}_2$ and $\mathcal{P}_3$ are disjoint, we could swap $\mathcal{P}_2$ with $\mathcal{P}_3$ to produce two new offspring $[\mathcal{P}_1, \mathcal{P}_3]$ and $[\mathcal{P}_2, \mathcal{P}_4]$.

5) Mutation: The objective of the mutation operation is to *diversify* the genes of the current population, which helps prevent the solution from being trapped in a local optimum. Just as some malicious mutations could happen in the natural world, mutation in GA may produce individuals that have worse fitness values. In such cases, some "filtering" operation is needed (e.g., the selection operation) to reject such "bad" genes and to drive GA toward optimality.

Mutation is performed on an individual with probability μ (called the *mutation rate*). For better performance, we propose a schedule to vary the mutation rate within $[\mu_{min}, \mu_{max}]$ over iterations (rather than using a fixed μ). The mutation rate is first initialized to μ_{max}; then as generation number k increases, the mutation rate gradually decreases to μ_{min}, i.e.,

$$\mu_k = \mu_{max} - \frac{k \cdot (\mu_{max} - \mu_{min})}{T_{max}} \tag{16}$$

where T_{max} is the maximum number of generations. Our results show that varying the mutation rates over generations significantly improves the online performance of the GA-based routing scheme. In essence, such schedule of μ is similar to the cooling schedule used in SA, and yields better convergence performance for Problem OPT-MM.

Fig. 7 illustrates a simple example of the mutation operation. In this example, we could implement mutation as follows. First, we choose a path $\mathcal{P}_h$, $h = 1$ or 2, with equal probabilities. Then, we can randomly pick an integer value k in the interval

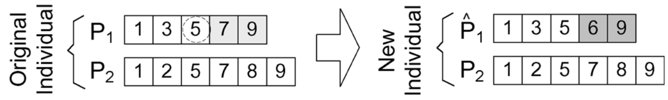


Fig. 7. Example of the mutation operation.

$[2, |\mathcal{P}_h| - 1]$, where $|\mathcal{P}_h|$ is the cardinality of $\mathcal{P}_h$, and let the partial path $\{s, \cdots, g_k\}$ be $\mathcal{P}_h^u$, where g_k is the k-th node along $\mathcal{P}_h$. Finally, we can use any constructive approach to build a partial path from g_k to t, denoted as $\mathcal{P}_h^d$, which does not repeat any node in $\mathcal{P}_h^u$ other than g_k. If no such alternative segment exists between g_k and t, we keep the path intact; otherwise, a new path can now be created by concatenating the two partial paths as $\mathcal{P}_h^u \cup \mathcal{P}_h^d$.

V. Simulation Results

In this section, we present our simulation studies on multimedia-centric routing. In each experiment, we generate a wireless ad hoc network topology by placing a number of nodes at random locations in a rectangular region, where connectivity is determined by the distance coverage of each node's transmitter. The area is adjusted for networks with different numbers of nodes in order to achieve an appropriate node density to provide a connected network. The source destination nodes s and t are uniformly chosen from the nodes.

A comprehensive simulation would require accurate models for the entire protocol stack, as well as accurate and efficient link statistic measurement schemes, which remains an area of active research [19], [20]. In order to simplify the simulations, we focus on the network layer and application layer models. For every link, the failure probability is uniformly chosen from [0.01, 0.3]; the available bandwidth is uniformly chosen from [100, 400] Kb/s, with 50 Kb/s steps; the mean burst length is uniformly chosen from [2], [6]. A DD video codec is implemented and used in the simulations. A practical distributed implementation architecture of the proposed scheme is presented in Section VI.

We set the GA's parameters as follows: the population size is 15; $\theta = 0.7$; μ is varied from 0.3 to 0.1 using the schedule described in Section IV; σ^2 is set to 1, since it does not affect path selection decisions. The GA is terminated after a predefined number of generations or after a prespecified computation time elapsed. The best individual found by the GA is prescribed as the solution to Problem OPT-MM.

A. Optimality of GA Solutions

One important performance concern is the quality of the GA solutions. As discussed, due to the complex nature of Problem OPT-MM, a closed-form optimal solution is not obtainable. However, for small networks, an optimal solution may be numerically obtained via an exhaustive search and can be used to compare with the proposed GA-based solutions.

Table II shows the optimal distortion values found by GA (each is the average of 30 runs) and by exhaustive search for two 10-node and two 15-node networks. We find that the solutions found by GA are very close to the global optimum in all cases. In addition, the standard deviation of the 30 GA results for the same network is negligibly small, indicating negligible confidence intervals. The average computational time for GA is 0.29 s for the 10-node network (about 60 generations) and 0.39 s for the 15-node network (about 70 generations) on a Pentium 4 2.4 GHz computer (512 MB memory) with MATLAB 6.5. For exhaustive search, the average computational time is 58.7 s for the 10-node case and 1877 s for the 15-node case.

TABLE II
Comparison of the Average Distortions Obtained by the GA-Based Routing and Exhaustive Search

	Topo. 1	Topo. 2	Topo. 3	Topo. 4
Network Size	10-node	10-node	15-node	15-node
Global Opt.	0.3308	0.2004	0.3863	0.2969
GA (average)	0.3330	0.2004	0.3937	0.2972
GA (std. dev.)	7.6e-6	0	2.8e-5	2.9e-6
Lower Bound	0.2810	0.1832	0.3527	0.2444

We also compute the lower bound using ALG-LB for each of the networks. The results are given in the last row of Table II. We observe that the lower bounds are within 8% to 16% percentile of the global optimum.

B. Comparison With Trajectory Methods

For comparison purposes, we implemented simulated annealing (SA) and tabu search (TS), both of which have been used in solving certain networking problems. We used the *geometric cooling schedule* for the SA implementation with a decay coefficient $\omega = 0.99$ [21]. For the TS implementation, we choose a *tabu list* of five for small networks and ten for large networks [22].

In Fig. 8, we plot the evolution of distortion values obtained by GA, SA, and TS for a 10-node network and a 50-node network, respectively. All the three metaheuristics are terminated after running for 1 s. Upon termination, GA has evolved 210 generations in Fig. 8(a) and 75 generations in Fig. 8(b); SA ran for 1500 iterations in Fig. 8(a) and 700 iterations in Fig. 8(b); TS ran for 1050 iterations in Fig. 8(a) and 550 generations in Fig. 8(b). GA has fewer number of iterations than SA and TS, due to its higher computational complexity (see Section VI-B). For both networks, the best distortion values found by GA are evidently much better than those by SA or TS. In Fig. 8(a), GA quickly converges to the global optimal, while both SA and TS are trapped at local optima (i.e., no further decrease in distortion value after hundreds of iterations). The same trend can be observed in the 50-node network case shown in Fig. 8(b), although the global optimum is not obtainable here.

An interesting observation from Fig. 8 is that for GA, the biggest improvement in distortion is achieved in the initial iterations, while the improvement gets smaller as GA evolves more generations. The initial population is generated using the random construction method discussed in Section IV-A, with no consideration on video performance. The initial solutions usually have high distortion values. The distortion value quickly drops over iterations, indicating that the GA performance is not very sensitive to the quality of the initial population. Also note that the SA and TS curves increase at some time instances [e.g, the TS curve at 0.06 s in Fig. 8(a) and the SA curve at 0.08 s

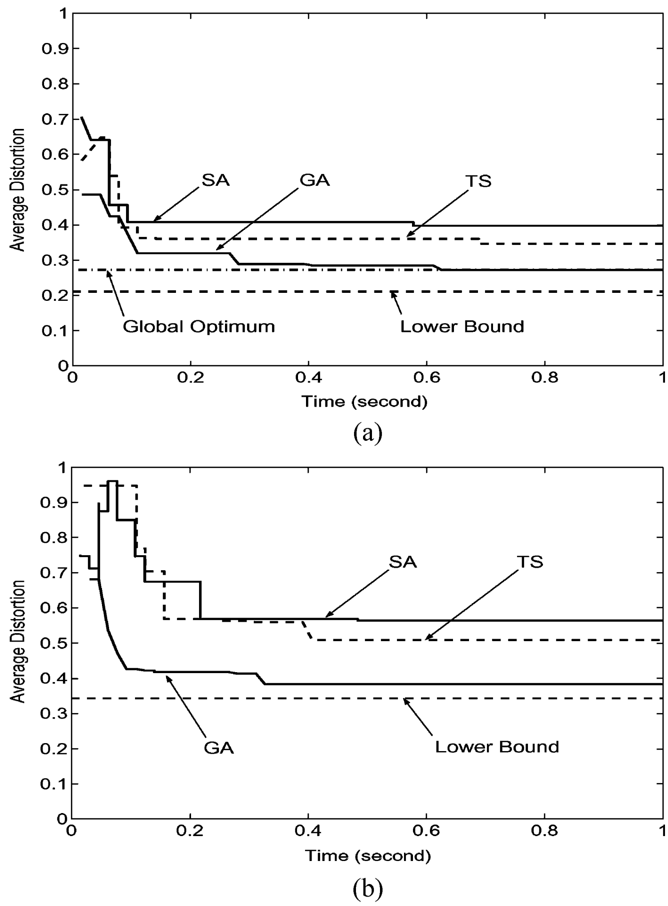


Fig. 8. Comparison of distortion evolution of three metaheuristic methods. (a) Distortion evolution for a 10-node network; (b) distortion evolution for a 50-node network.

in Fig. 8(b)], which implies that a nonimproving solution is accepted in order to escape from local minima.

We also plot the lower bounds derived using ALG-LB in the figures, which are quite tight in both cases. In addition to providing much better solutions, another strength of GA over trajectory methods is that multiple "good" solutions can be found after a single run. Such extra good paths can be used as alternative (or backup) paths if needed.

C. Comparison With Network-Centric Approaches

In this section, we compare our GA approach with network-centric routing approaches. Although there are many alternative approaches, we implement two popular network-centric multipath routing algorithms, namely k-shortest path (SP) routing (with $k = 2$ or 2-SP) [23] and disjoint path routing, Disjoint Pathset Selection Protocol (DPSP) [24]. Our 2-SP implementation uses hop count as routing metric such that two shortest paths are found. In our DPSP implementation, we set the link costs to $\log(1/p_{ij})$, for all $\{i,j\} \in E$, such that two disjoint paths having the highest end-to-end success probabilities are found. We compare the performance of our GA-based multipath routing with these two algorithms over a 50-node ad hoc network using a video clip.

There are many ways to generate MD video (see [1] for an excellent survey). We choose a time-domain partitioning coding

TABLE III
COMPARISON OF GA AND NETWORK-CENTRIC ROUTING

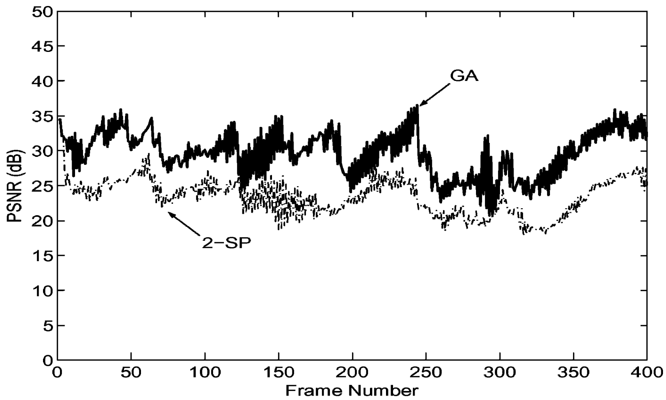
Routing Algorithm	$\mathcal{P}_1$ Succ. Ratio	$\mathcal{P}_2$ Succ. Ratio	Desc. 1 Rate	Desc. 2 Rate	Average PSNR
GA	0.994	0.952	350 Kb/s	350 Kb/s	29.71 dB
2-SP	0.798	0.782	100 Kb/s	200 Kb/s	23.42 dB
DPSP	0.965	0.793	100 Kb/s	100 Kb/s	25.65 dB

scheme, where two descriptions are generated by separating the even and odd-numbered frames and coding them separately. This simple time-domain partitioning method is widely used in many video streaming studies [2], [4]–[6]. Compared with a traditional single description coder, this coder has a comparable computational complexity. Its coding efficiency is slightly lower than a single description coder, due to the fact that a longer motion prediction distance is used. However, this reduced coding efficiency is well justified by the resulting enhanced error resilience. We use an H.263+ like codec. Since our approach is quite general, we conjecture that the same trend in performance would be observed for other video codecs, such as H.264 or MPEG-2 or MPEG-4. The QCIF sequence "Foreman" (400 frames) is encoded at 15 fps for each description. A 10% macroblock level intra-refreshment is used. Each Group of Blocks (GOB) is carried in a different packet. The received descriptions are decoded and PSNR values of the reconstructed frames computed. When a GOB is corrupted, the decoder applies a simple error concealment scheme by copying from the corresponding slice in the most recent, correctly received frame.

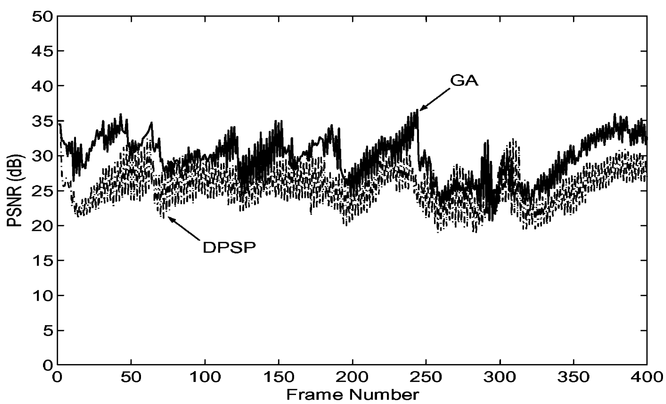
The quality of the paths found by the three algorithms are presented in Table III. The 2-SP algorithm has the worst performance in terms of path success probabilities. The DPSP algorithm has an improved success probability performance since it uses link success probabilities in routing. However, it may sacrifice path bandwidth while pursuing low loss paths. As a result, it produces the lowest end-to-end bandwidths. We observe that our GA-based routing yields paths with much higher end-to-end success probabilities and end-to-end bandwidths, resulting in greatly improved video quality.

The PSNR curves of the received video frames are plotted in Fig. 9. We observe that the PSNR curve obtained by GA is well above those obtained by the aforementioned network-centric routing approaches. Using GA, the improvements in average PSNR value over 2-SP and DPSP are 6.29 dB and 4.06 dB, respectively. We also experiment with an improved 2-SP algorithm which also uses link success probabilities as link metric (as in DPSP). In this case, our GA-based routing achieves a 1.27 dB improvement over this enhanced 2-SP version, which is still significant in terms of visual video quality.

An inherent issue of transmitting video over multiple paths is that when the paths are unbalanced, e.g., either in bandwidth or in loss characteristics, the streams may have different qualities. When interlaced and displayed, such unbalanced streams may cause large variations in frame quality and yield low subjective quality (although a high objective quality, e.g., average PSNR, may always be achieved). In Problem OPT-MM, due to



(a)



(b)

Fig. 9. PSNR curves of received video sequences. (a) GA-based algorithm versus 2-SP; (b) GA-based algorithm versus DPSP.

the symmetry in (1), our GA-based routing attempts to find a balanced pair of paths while minimizing D. For example, the two paths found by GA as in Table III have similar success probabilities and exactly the same bandwidth, resulting in relatively balanced descriptions. In the case when the descriptions are highly unbalanced, the problem can be further alleviated by using an advanced MD coder that is capable of producing descriptions with unbalanced rates (but with relatively equal qualities) or by striping packets of the descriptions across multiple paths to make losses of the descriptions relatively even.

VI. Distributed Implementation

In addition to a centralized version of the GA-based multipath routing, we also consider how to develop an effective distributed implementation for practical systems. Our approach is to implement our cross-layer routing algorithms by incorporating some proven ideas from existing network layer ad hoc routing algorithms.

A. A Distributed Implementation Architecture

We believe that the GA-based routing is most suitable to be implemented within the proactive ad hoc routing paradigm. Our choice is motivated by the following two important and practical considerations. First, it is necessary to make quick routing decisions whenever a new MD video request arrives. The readily available route information under a proactive paradigm is well suited for this purpose. Second, for many applications (e.g.,

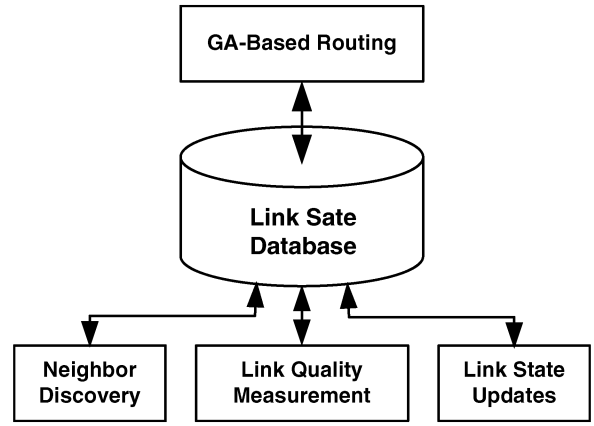


Fig. 10. Distributed implementation architecture for the GA-based multipath routing.

search and rescue), it is highly desirable to maintain an accurate network topology and link state information at an ad hoc node for administrative purposes.

At the core of distributed implementation are efficient means to build and maintain network topology and link statistics databases at each node. To this end, we find that the class of link state routing protocols, such as the *Optimized Link State Routing* protocol (OLSR) [10] and *Topology Dissemination Based on Reverse-Path Forwarding* (TBRPF) [25], are very suitable for this purpose. Fig. 10 depicts an implementation architecture of the proposed GA-based multipath routing at an ad hoc node. This implementation works in a completely distributed manner, and thus does not depend on any central entity in the network. We briefly describe the operations of its key modules.

1) Neighbor Discovery and Link Quality Measurement: Each node should detect its neighbor nodes to which a wireless link exists. This can be accomplished by periodically broadcasting HELLO messages containing information about neighbors and their link status to its one-hop neighbors. Each node continuously measures the link metrics, such as bandwidth, loss rate, and delay. Note that the measurements should be smoothed with a time window, rather than using instantaneous values. Several effective algorithms for such measurements (e.g., those proposed in [19]) can be used for this purpose.

2) Link State Updates: As with other link state routing protocols, our implementation also periodically broadcasts link state advertisements (LSA) to the entire network to distribute network topology information and link statistics. In order to reduce the control traffic overhead, we can use the *MultiPoint Relay* (MPR) technique [10] to minimize the flooding of control messages. It has been shown that MPR can effectively suppress control traffic overhead. Furthermore, the denser and larger a network is, the more reduction can be achieved [10].

3) Link State Database: Network topology and link statistics learnt from received LSAs are pooled in a *link state database*. Each link item (along with its corresponding metrics) is associated with a sequence number, which is set to the sequence number of the LSA message from which this link item was learned. Therefore, a stale item will be overwritten by a fresher item, making the link state database consistent and up-to-date.

4) GA-Based Multipath Routing: We can then apply the GA-based multipath routing to compute a set of routes from this node to other nodes in the network, as discussed in the previous sections. When the paths are computed, the next question is how to establish the paths for a video session. This could be done with *source routing*, within which each data packet carries the entire path in the packet header [9]. Alternatively, a technique such as *soft flow states* [26] can be used. With soft flow states, only the first packet contains the full path information. As the first packet travels from source to destination, the flow state mechanism allows each intermediate node to record the address of the next hop along this source route. Subsequent packets from the same flow may then be forwarded along the same route without the need to carry the same source routing information in the packet. Such per-flow state will be refreshed by a new packet belonging to the same flow, and will expire after a timeout period.

During the video session, the receiver keeps on monitoring the received video quality. As the network topology changes (e.g., due to mobility), the received video quality may fall below a threshold. Upon detecting such events, the receiver will notify the sender (via feedback, e.g, RTCP's Receiver Reports) and the sender will start the routing algorithm to find new paths.

B. Performance Issues

A practical consideration for the GA-based approach is computational complexity. Since GA evolves a population of solutions, it has higher computational complexity than trajectory methods or network-centric multipath routing schemes. We conjecture that such complexity should not be an issue for most ad hoc networks with laptop nodes or even workstation nodes (e.g., carried in military vehicles), and wireless mesh networks where the wireless routers are usually computationally powerful and plugged in power outlets [27]. Our numerical results show that a properly designed GA can compute very good routes for ad hoc networks with small and moderate sizes (e.g., 50 nodes as in Fig. 8) in several hundred milliseconds, which are fast enough for practical uses. Furthermore, our numerical results show that with GA, the greatest improvement in fitness value is achieved after a small number of generations, and the improvement gets much smaller after these initial generations (see Fig. 8). Therefore, there is a tradeoff between solution optimality and computation time. For a delay-sensitive real-time application (or low-end devices), GA can compute a set of "good" routes very quickly, and the application can use these "good" routes after a very small delay. As GA continues to evolve, the routes used by the application can be updated with newly computed better set of routes for enhanced performance.

It is also worth noting that the general framework presented in the paper could be adapted to support various applications. For example, for stored video, the description rates are fixed. As a result, d_0, d_1 and d_2 in (10) will be constants. For live video, the encoder could also fixes its encoding rates, and routing could be performed with the fixed rates in order to reduce complexity. These are all application-specific options.

VII. Related Work

Multipath routing has been an active research area over the years. Various algorithms have been proposed to compute k-shortest paths [23], node- or link-disjoint paths [16], [24], or braided multiple paths [28]. Given multiple paths, traffic proportioning schemes are also designed to disperse traffic to the paths for an increased end-to-end throughput, load balancing, and fast failure recovery [29]. In the area of wireless ad hoc networks, many existing routing protocols are multipath-capable. However, most of these multipath routing algorithms are *network-centric*: they do not explicitly address application layer performance issues from a cross-layer perspective, as illustrated in Section V.

The problem of multipath routing for video has recently been explored in [4], [6]. Several MD surrogate selection algorithms were presented in [4], which provide guidelines on selecting MD video servers in a content delivery network. Although MD servers are selected such that video distortion is minimized, the problem of finding the optimal routes to the servers has not been explicitly addressed in this work. In [6], Begen *et al.* studied the multipath routing problem in the context of overlay networks, where path selection is formulated as an optimization problem that minimizes video distortion. However, the formulated problem is actually solved by an *exhaustive search* over the exponential solution space.

The potential of GA in addressing networking problems has been recognized in recent years. For example, GA has been explored to address various networking problems such as routing [18], [30], [31], admission control [32], channel assignment [33], network design [34], scheduling [35] and buffer management [36]. These efforts have made the important step in exploring the potential of GA for network optimization. The research in this paper builds upon these efforts and aims to address the more complex *cross-layer* optimization problem, which requires not only knowledge at the network layer, but also a deep understanding at the application layer in order to fully exploit the design and optimization space across the layers.

VIII. Conclusions

In this paper, we investigated the problem of optimal multipath routing for MD video. We found that a GA-based approach is eminently suitable to address such optimal routing problems, which involve complex objective functions and exponential solution spaces. We consequently designed a GA-based algorithm to address this multipath routing problem and found that this approach provides near-optimal results. We also developed a tight lower bound for distortion, which can be used to evaluate the performance of a GA-based solution as well as to establish its termination criteria. Although a GA-based approach is a centralized algorithm in nature, we showed that it is amenable for distributed implementation. Currently, we are developing a multipath video testbed to implement and demonstrate these ideas.

APPENDIX I
PROOFS

A. Proof of Property M3

For solution $\hat{x} = \{\mathcal{P}_1, \mathcal{P}_2\}$ in Fig. 2(a), let there be K joint links with parameters $\{\alpha_k, \beta_k\}$, $k = 1, \cdots, K$. Let $a = 2^{-2R_1}$ and $b = 2^{-2R_2}$. Then we can derive the difference in the two distortion values as

$$\begin{aligned}\frac{D(\hat{x}) - D(\bar{x})}{\sigma^2} &= p_{jnt} \cdot p_{dj}^1 \cdot p_{dj}^2 \cdot \frac{(a+b)(1-a)(1-b)}{a+b(1-a)} \\ &\quad \cdot \prod_{k=1}^{K-1} (1-\alpha_k)(1-p_K)(1-\alpha_K-\beta_K) \\ &\geq 0\end{aligned}$$

according to the "bursty" assumption in Property M3.

B. Proof of Proposition 1

The formation of the optimal solution $x^* = \{\mathcal{P}_1^*, \mathcal{P}_2^*\}$ could conform with one of the following two cases.

Case I: x^* is comprised of a pair of disjoint paths. From the construction procedure, x_l^* offers higher description rates than x^*, since b^* is optimal over all feasible paths. In addition, x_l^* offers higher end-to-end success probabilities for the descriptions than x^*, since p^* is optimal over all feasible paths. Then, we have that $D(x_l^*) \leq D(x^*)$ according to properties M1 and M2.

Case II: x^* is comprised of a pair of joint paths. Assume $\mathcal{P}_1^*$ and $\mathcal{P}_2^*$ share K links. Firstly, we construct a virtual solution $\bar{x} = \{\bar{\mathcal{P}}_1, \bar{\mathcal{P}}_2\}$, by i) appending a copy of each shared link k to the disjoint portions of the two paths; and ii) removing link k from the shared portion, $k = 1, \cdots, K$. That is, the resulting $\bar{\mathcal{P}}_h$ has the same set of links as $\mathcal{P}_h^*$, $h = 1, 2$, but $\bar{\mathcal{P}}_1$ is completely disjoint with $\bar{\mathcal{P}}_2$. As a result, x_l^* may not be feasible; an example of such construction operation is shown in Fig. 2.

Secondly, we observe that the constructed solution $\bar{x}$ can provide a pair of description rates no lower than x^*, since the K links originally shared by the two descriptions are now used exclusively by each description. By applying Property M3 iteratively for K times, we have that $D(\bar{x}) \leq D(x^*)$.

Thirdly, from ALG-LB, we have that $b^* \geq B(\bar{\mathcal{P}}_h)$, $h = 1, 2$ (recall that $B(\mathcal{P})$ is the end-to-end bandwidth of path $\mathcal{P}$). Therefore, x_l^* dominates the constructed $\bar{x}$ in terms of end-to-end bandwidths (i.e., $b^* \geq B(\bar{\mathcal{P}}_h)$, $h = 1, 2$). Similarly, we can also show that x_l^* dominates $\bar{x}$ in terms of end-to-end success probabilities (i.e., $p^* \geq p_{dj}^h(\bar{\mathcal{P}}_h)$, $h = 1, 2$). According to properties M1 and M2, we have that $D(x_l^*) \leq D(\bar{x})$.

Finally, we have that $D(x_l^*) \leq D(\bar{x}^*) \leq D(x^*)$.

REFERENCES

[1] V. Goyal, "Multiple description coding: compression meets the network," *IEEE Signal Process. Mag.*, vol. 18, pp. 74–93, Sep. 2001.

[2] S. Mao, S. Lin, S. Panwar, Y. Wang, and E. Celebi, "Video transport over ad hoc networks: multistream coding with multipath transport," *IEEE J. Sel. Areas Commun.*, vol. 12, no. 10, pp. 1721–1737, Dec. 2003.

[3] N. Gogate, D. Chung, S. Panwar, and Y. Wang, "Supporting image/video applications in a multihop radio environment using route diversity and multiple description coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, no. 9, pp. 777–792, Sep. 2002.

[4] J. Apostolopoulos, T. Wong, W. Tan, and S. Wee, "On multiple description streaming in content delivery networks," in *Proc. IEEE INFOCOM*, New York, Jun. 2002, pp. 1736–1745.

[5] J. Chakareski, S. Han, and B. Girod, "Layered coding vs. multiple descriptions for video streaming over multiple paths," in *Proc. ACM Multimedia*, Berkeley, CA, Nov. 2003, pp. 422–431.

[6] A. Begen, Y. Altunbasak, O. Ergun, and M. Ammar, "Multi-path selection for multiple description encoded video streaming," *EURASIP Signal Process.: Image Commun.*, vol. 20, no. 1, pp. 39–60, Jan. 2005.

[7] T. Back, D. Fogel, and Z. Michalewicz, Eds., *Handbook of Evolutionary Computation*. New York: Oxford Univ. Press, 1997.

[8] C. Blum and A. Roli, "Metaheuristics in combinatorial optimization: overview and conceptual comparison," *ACM Comput. Surv.*, vol. 35, no. 3, pp. 268–308, Sep. 2003.

[9] D. Johnson, D. Maltz, and Y.-C. Hu, The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks (DSR) Apr. 2003, IETF Internet Draft.

[10] T. Clausen and P. Jacquet, Optimized Link State Routing Protocol Oct. 2003, IETF RFC 3626.

[11] S. Mao, S. Kompella, Y. Hou, H. Sherali, and S. Midkiff, "Routing for concurrent video sessions in ad hoc networks," *IEEE Trans. Veh. Technol.*, vol. 55, no. 1, Jan. 2006.

[12] E. Setton, Y. Liang, and B. Girod, "Adaptive multiple description video streaming over multiple channels with active probing," in *Proc. IEEE ICME*, Baltimore, MD, Jul. 2003.

[13] L. Ozarow, "On a source coding problem with two channels and three receivers," *Bell Syst. Tech. J.*, vol. 59, no. 10, pp. 84–91, Dec. 1980.

[14] M. Alasti, K. Sayrafian-Pour, A. Ephremides, and N. Farvardin, "Multiple description coding in networks with congestion problem," *IEEE Trans. Inform. Theory*, vol. 47, no. 3, pp. 891–902, Mar. 2001.

[15] D. Fritchman, "A binary channel characterization using partitioned markov chains," *IEEE Trans. Inform. Theory*, vol. 13, no. 2, pp. 221–227, Apr. 1967.

[16] H. Sherali, K. Ozbay, and S. Subramanian, "The time-dependent shortest pair of disjoint paths problem: complexity, models, and algorithms," *Networks*, vol. 31, no. 4, pp. 259–272, Dec. 1998.

[17] N. Malpani and J. Chen, "A note on practical construction of maximum bandwidth paths," *Inform. Process. Lett.*, vol. 83, no. 3, pp. 175–180, Aug. 2002.

[18] C. Ahn and R. Ramakrishna, "A genetic algorithm for shortest path routing problem and the sizing of populations," *IEEE Trans. Evol. Comput.*, vol. 6, no. 6, pp. 566–579, Dec. 2002.

[19] The QoS With the OLSR Protocol Homepage [Online]. Available: http://qolsr.lri.fr/

[20] R. Kapoor, L.-J. Chen, L. Lao, M. Gerla, and M. Sanadidi, "Capprobe: a simple and accurate capacity estimation technique," in *Proc. ACM SIGCOMM*, Portland, OR, Oct. 2004, pp. 67–78.

[21] E. Aarts and J. Korst, *Simulated Annealing and Boltzman Machines*. New York: Wiley, 1989.

[22] F. Glover and M. Laguna, *Tabu Search*. Boston, MA: Kluwer, 1997.

[23] D. Eppstein, "Finding the k shortest paths," *SIAM J. Comput.*, vol. 28, no. 2, pp. 652–673, Aug. 1999.

[24] P. Papadimitratos, Z. Haas, and E. Sirer, "Path set selection in mobile ad hoc networks," in *Proc. ACM Mobihoc*, Lausanne, Switzerland, Jun. 2002, pp. 1–11.

[25] R. Ogier, F. Templin, and M. Lewis, Topology Dissemination Based on Reverse-Path Forwarding (TBRPF) Feb. 2004, IETF RFC 3684.

[26] Y.-C. Hu and D. Johnson, "Design and demonstration of live audio and video over multihop wireless ad hoc networks," in *Proc. IEEE Milcom*, Anaheim, CA, Oct. 2002, pp. 7–10.

[27] I. Akyildiz and X. Wang, "A survey on wireless mesh networks," *IEEE Commun. Mag.*, vol. 43, no. 9, Sep. 2005.

[28] S. Murthy and J. Garcia-Luna-Aceves, "Congestion-oriented shortest multipath routing," in *Proc. IEEE INFOCOM*, San Francisco, CA, May 1996, pp. 1038–1036.

[29] E. Gustafsson and G. Karlsson, "A literature survey on traffic dispersion," *IEEE Network*, vol. 11, no. 2, pp. 28–36, Mar. 1997.

[30] N. Banerjee and S. Das, "Fast determination of QoS-based multicast routes in wireless networks using genetic algorithm," in *Proc. IEEE ICC*, Jun. 2001, pp. 2588–2592.

[31] M. Sinclair, "Minimum cost wavelength-path routing and wavelength allocation using a genetic-algorithm/heuristic hybrid approach," *IEE Proc. Commun.*, vol. 46, no. 1, pp. 1–7, Feb. 1999.

[32] A. Yener and C. Rose, "Genetic algorithms applied to cellular call admission: local policies," *IEEE Trans. Veh. Technol.*, vol. 46, no. 1, pp. 72–79, Feb. 1997.

[33] S. Ghosh, B. Sinha, and N. Das, "Channel assignment using genetic algorithm based on geometric symmetry," *IEEE Trans. Veh. Technol.*, vol. 52, no. 4, pp. 860–875, Jul. 2003.

[34] R. Elbaum and M. Sidi, "Topological design of local-area networks using genetic algorithms," *IEEE/ACM Trans. Networking*, vol. 4, no. 5, pp. 766–778, Oct. 1996.

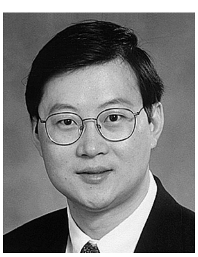
[35] C. Ngo and V. Li, "Centralized broadcast scheduling in packet radio networks via genetic-fix algorithms," *IEEE Trans. Commun.*, vol. 51, no. 9, pp. 1439–1441, Sep. 2003.

[36] G. Fatta, F. Hoffmann, G. Re, and A. Urso, "A genetic algorithm for the design of a fuzzy controller for active queue management," *IEEE Trans. Syst., Man, Cybern. C*, vol. 33, no. 3, pp. 313–324, Aug. 2003.

Shiwen Mao (S'99–M'04) received the B.S. and the M.S. degree from Tsinghua University, Beijing, China, in 1994 and 1997, respectively, both in electrical engineering. He received the M.S. degree in system engineering and the Ph.D. degree in electrical and computer engineering from Polytechnic University, Brooklyn, NY, in 2000 and 2004, respectively.

He was a Research Member at IBM China Research Lab, Beijing, from 1997 to 1998. In the summer of 2001, he was a research intern at Avaya Labs-Research, Holmdel, NJ. Currently, he is a Research Scientist in the Department of Electrical and Computer Engineering, Virginia Polytechnic Institute and State University, Blacksburg. His research interests include cross-layer design and optimization, wireless ad hoc and sensor networks, and multimedia networking.

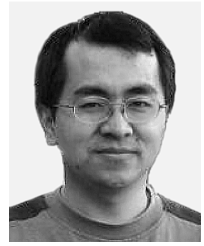
Dr. Mao is a co-recipient of the 2004 IEEE Communications Society Leonard G. Abraham Prize in the Field of Communications Systems. He is the co-author of a textbook *TCP/IP Essentials: A Lab-Based Approach* (Cambridge University Press, 2004).



Y. Thomas Hou (S'91–M'98–SM'04) received the B.E. degree from the City College of New York in 1991, the M.S. degree from Columbia University, New York, in 1993, and the Ph.D. degree from Polytechnic University, Brooklyn, NY, in 1998, all in electrical engineering.

From 1997 to 2002, he was a Research Scientist and Project Leader at Fujitsu Laboratories of America, IP Networking Research Department, Sunnyvale, CA. Since Fall 2002, he has been an Assistant Professor with the Bradley Department of Electrical and Computer Engineering, Virginia Polytechnic Institute and State University, Blacksburg. His research interests are in the algorithmic design and optimization for network systems. His current research focuses on wireless sensor networks and multimedia over wireless ad hoc networks. In recent years, he has worked on scalable architectures, protocols, and implementations for differentiated services Internet; service overlay networking; multimedia streaming over the Internet; and network bandwidth allocation policies and distributed flow control algorithms. He has published extensively in the above areas.

Dr. Hou is a co-recipient of the 2004 IEEE Communications Society Multimedia Communications Best Paper Award, the 2002 *IEEE International Conference on Network Protocols* (ICNP) Best Paper Award, and the 2001 IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY (CSVT) Best Paper Award.



Xiaolin Cheng (S'04) received the B.E. and M.E. degrees in automation from Tsinghua University, China, in 1997 and 2000 respectively, and the M.S. in computer engineering from the Virginia Polytechnic Institute and State University, Blacksburg, in 2005. Currently, he is pursuing the Ph.D. degree in the Computer Science Department at the University of California-Davis.

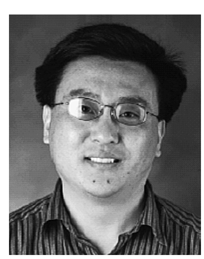
He was a Senior Engineer at Panasonic Beijing Labs for two-and-a-half years. His research focuses on multimedia delivery over wireless networks and routing algorithms for wireless ad hoc networks.

Hanif D. Sherali is W. Thomas Rice Endowed Chaired Professor of Engineering in the Industrial and Systems Engineering Department at the Virginia Polytechnic Institute and State University, Blacksburg. His area of research interest is in discrete and continuous optimization, with applications to location, transportation, and engineering design problems. He has published about 200 papers in operations research journals, has co-authored four books in this area, and serves on the editorial board of eight journals.

Dr. Sherali is a Fellow of the IIE, a Fellow of INFORMS, and a member of the National Academy of Engineering.

Scott F. Midkiff (S'82–M'85–SM'92) joined the Bradley Department of Electrical and Computer Engineering at Virginia Tech in 1986 and is now a professor. He previously worked at Bell Laboratories and held a visiting position at Carnegie Mellon University. He received his B.S.E. and Ph.D. from Duke University and his M.S. from Stanford University, all in electrical engineering.

His research interests include system issues in wireless and ad hoc networks, network services for pervasive computing, and performance modeling of mobile ad hoc networks.



Ya-Qing Zhang (S'87–M'90–SM'93–F'97) is the Corporate Vice President of Microsoft Corporation's Mobile and Embedded Devices Division, Seattle, WA. He was previously the Managing Director of Microsoft Research Asia, Beijing, China, and the Director of the Multimedia Technology Laboratory, Sarnoff Corporation, Princeton, NJ (formerly David Sarnoff Research Center and RCA Laboratories). He was with GTE Laboratories, Inc., Waltham, MA, and Contel Technology Center, VA, from 1989 to 1994. He has been engaged in research and commercialization of MPEG2/DTV, MPEG4/VLBR, and multimedia information technologies. He has more than 70 U.S. patents granted or pending and has authored or contributed to more than a dozen books and 300 journal and conference articles.

Dr. Zhang served as the Editor-in-Chief for the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY from July 1997 to July 1999. He also serves on the editorial boards of a number of leading professional journals.