Multipath Routing for Multiple Description Video in Wireless Ad Hoc Networks

Shiwen Mao† Y. Thomas Hou† Xiaolin Cheng† Hanif D. Sherali‡ Scott F. Midkiff†

†The Bradley Department of Electrical and Computer Engineering
‡The Grado Department of Industrial and Systems Engineering
Virginia Tech, Blacksburg, VA 24061
{smao, thou, xiaolin, hanifs, midkiff}@vt.edu

***Abstract*— As developments in wireless ad hoc networks continue, there is an increasing expectation with regard to supporting content-rich multimedia communications (e.g., video) in such networks, in addition to simple data communications. The recent advances in *multiple description* (MD) video coding have made it highly suitable for multimedia applications in such networks. In this paper, we study the important problem of multipath routing for MD 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, as well as a tight lower bound for video distortion. We use numerical results to demonstrate the superior performance of the GA-based approach and compare it to several other approaches. Our efforts in this work provide an important methodology for addressing complex cross-layer optimization problems, particularly those involving application and network layers.**

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 controlled 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. This new video coding technique is quite different from traditional single stream video coding, where received video quality is highly susceptible to the dynamics of a single path. MD coding is also drastically different from layered video coding, where the successful video decoding depends on the base layer. 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 most of the paths in such networks are highly 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 for most of the streaming period, while video quality improves as more descriptions are received.

Several researchers have proposed to use MD coding with *multipath routing* for multimedia transport [2]–[6]. These efforts 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 the best paths for the descriptions has not been adequately addressed. In a recent work in [7], Begen *et al.* studied the problem of multipath routing for MD video in the context of Internet overlay networks. The optimal routing problem is, however, solved via exhaustive search which has an exponential complexity. To reduce the computational complexity, a heuristic algorithm was proposed in [8]. However, this heuristic relies on the special hierarchical structure of the Internet and overlay networks, and may not be suitable for wireless ad hoc networks.

In this paper, we study the important problem of multipath

0-7803-8968-9/05/$20.00 (C)2005 IEEE

routing for MD 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 polynomials and non-decomposable. Consequently, it would be futile to develop a tractable analytic solution. However, we find that a metaheuristic technique such as *Genetic Algorithms* (GAs) [9] 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 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) [10]). 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 yet 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, particularly the class of *proactive* protocols [11], [12]. As an example, we present a distributed implementation based on the Optimized Link State Routing Protocol (OLSR) [11].

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

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 is 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 (or even a wireless link [13]) with consideration of mobility, interference, and the time-varying wireless channels is extremely difficult and remains active research. 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 $\{i, j\}$;
- p_{ij}: the probability when link $\{i, j\}$ is "up";
- l_{ij}: average burst length for packet losses on link $\{i, j\}$.

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} :	"up" 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 (6)
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 :	crossover rate
μ :	mutation rate

In practice, these parameters can be measured by every node, and distributed throughout the network using Link State Advertisements (LSA) [11]. 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 can be incorporated into this framework as well (e.g., see [14]). Table I lists the notation used in this paper.

A. Rate-Distortion Regions for MD 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]–[8]. 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 [15] demonstrates that the most significant performance gain is achieved when the number of descriptions increases from 1 to 2, with only marginal improvements achieved for further increases in number of descriptions.

For video coding and communications, a rate distortion model describes the relationship between the bit rate and the achieved distortion. For two descriptions (each generated for a sequence of video frames), denote d_h the achieved distortion when only Description h is received, $h = 1, 2$, and d_0 the distortion when both descriptions are received. Denote R_h the rate in bits/sample 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 [16]. For computational efficiency, Alasti *et al.* in [17] introduced the following rate-distortion region (also employed 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. Denote P_{00} 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 expressed as:

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

B. Description Rates and Success Probabilities

As a first step to formulate the problem of optimal multi-path 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 Eqs. (1) and (2)).

For a source-destination pair $\{s, t\}$, suppose we have two given paths $[\mathcal{P}_1, \mathcal{P}_2]$ in $\mathcal{G}\{V, E\}$. Since we do not mandate "disjointedness" in path selection, $\mathcal{P}_1$ and $\mathcal{P}_2$ may share nodes and links in $\mathcal{G}\{V, E\}$. Similar to [7], 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 consist 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. 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.

We now focus on how to compute the end-to-end success probabilities. We model each link $\{i, j\}$ as an on-off process modulated by a discrete-time Markov chain, as shown in

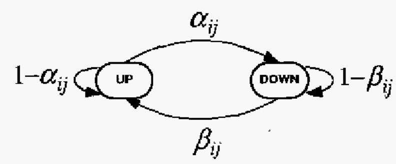


Fig. 1. The Gilbert two-state link model.

Figure 1. There is no packet loss when the link is in the "up" state; packet loss rate is 1 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 $\alpha_{ij} = (1 - p_{ij})/(p_{ij} l_{ij})$. For disjoint portion of the paths, it suffices to model the packet loss as a Bernoulli event, since losses on the two descriptions are assumed to be independent. 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. 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 an approach similar to [7] 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. Denote T_{on} the average length of the "up' period. Then,

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

The average success probability of the joint portion is:

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

Finally, the transition probabilities of the aggregate on-off process are:

$$\begin{cases} \alpha = \frac{1}{T_{on}} \\ \beta = \frac{p_{jnt}}{T_{on}(1 - p_{jnt})}. \end{cases}$$

Note that $\alpha = 0$ and $\beta = 0$ if $\mathcal{J}(\mathcal{P}_1, \mathcal{P}_2) = \emptyset$. The consolidated path model is illustrated in Figure 2, 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 p_{dj}^h, $h = 1, 2$.

With the above path model, the joint probabilities of receiving the descriptions can be computed as:

$$\begin{cases} P_{00} = p_{jnt} \cdot (1 - \alpha) \cdot p_{dj}^1 \cdot p_{dj}^2 \\ P_{01} = p_{jnt} \cdot p_{dj}^1 \cdot \left[1 - (1 - \alpha) \cdot p_{dj}^2\right] \\ P_{10} = p_{jnt} \cdot \left[1 - (1 - \alpha) p_{dj}^1\right] \cdot p_{dj}^2 \\ P_{11} = 1 - p_{jnt} \cdot \left[p_{dj}^1 + p_{dj}^2 - (1 - \alpha) \cdot p_{dj}^1 \cdot p_{dj}^2\right]. \end{cases} \tag{5}$$

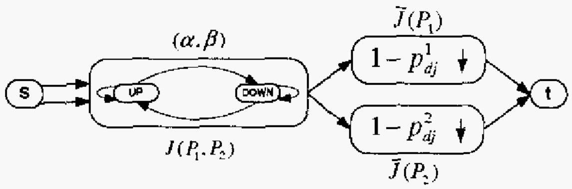
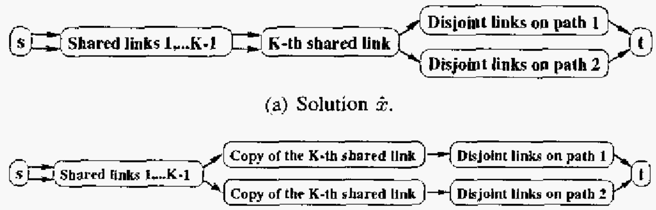


Fig. 2. A simplified path model for double-description video.



(a) Solution $\hat{x}$.

(b) Solution $\bar{x}$.

Fig. 3. 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 is appended to each of the disjoint portions).

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:

$$x_{ij}^h = \begin{cases} 1, & \text{if } \{i,j\} \in \mathcal{P}_h \\ 0, & \text{otherwise.} \end{cases} \tag{6}$$

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 Given a wireless ad hoc network $\mathcal{G}\{V,E\}$ and a source destination pair s–t,

Minimize: D (7)

subject to:

$$\sum_{j:\{i,j\}\in E} x_{ij}^h - \sum_{j:\{j,i\}\in E} x_{ji}^h = \begin{cases} 1, & \text{if } i=s, & \forall i \in V, h=1,2 \\ -1, & \text{if } i=t, & \forall i \in V, h=1,2 \\ 0, & \text{otherwise}, & \forall i \in V, h=1,2 \end{cases} \tag{8}$$

$$\sum_{j:\{i,j\}\in E} x_{ij}^h \begin{cases} \leq 1, & \text{if } i \neq t, & \forall i \in V, h=1,2 \\ =0, & \text{if } i=t, & \forall i \in V, h=1,2 \end{cases} \tag{9}$$

$$x_{ij}^1 \cdot R_1 + x_{ij}^2 \cdot R_2 \leq \rho \cdot b_{ij}, \quad \forall \{i,j\} \in E \tag{10}$$

$$x_{ij}^h \in \{0,1\}, \quad \forall \{i,j\} \in E, h=1,2. \tag{11}$$

In Problem OPT-MM, $\{x_{ij}^h\}$ are binary optimization variables. Constraints (8) and (9) guarantee that the paths are loop-free, while constraint (10) 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 (5).

Clearly, the objective function (7) is a highly complex ratio of high-order polynomials 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 [18], 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 non-decreasing (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 very useful in measuring the performance of a heuristic algorithm, as well as serving as a reference for setting its termination conditions.

We find that the average video distortion D possesses the following *monotonicity* properties (the proofs are presented in [19]):

M1: D is non-increasing with R_h, $h = 1,2$.

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

M3: Consider the two solutions $\hat{x}$ and $\bar{x}$ shown in Figure 3. 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 $D(\hat{x}) \geq D(\bar{x})$.

The assumption in Property M3 relates to the covariance of two consecutive failure events X_k and X_{k+1} on link $\{i,j\}$:

$$\mathrm{Cov}\{X_k, X_{k+1}\} = \frac{\alpha_{ij}\beta_{ij}}{(\alpha_{ij}+\beta_{ij})^2}(1 - \alpha_{ij} - \beta_{ij}). \tag{12}$$

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 Figure 3, if the Kth shared link has bursty or random losses, then $\bar{x}$ yields a distortion no higher than $\hat{x}$.

We are now ready to construct a simple yet tight lower bound on the average video distortion. Algorithm ALG-LB in

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 $x_l^* = \left\{\mathcal{P}_1^l, \mathcal{P}_2^l\right\}$, 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. 4. ALG-LB: a procedure to construct a solution x_l^* that yields a lower bound for distortion D.

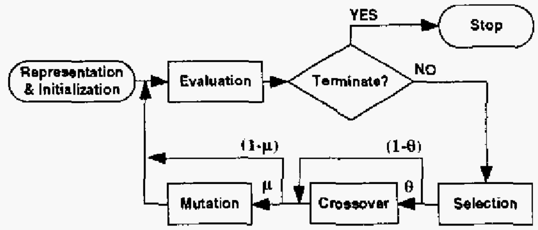


Fig. 5. Flow chart of our GA-based approach.

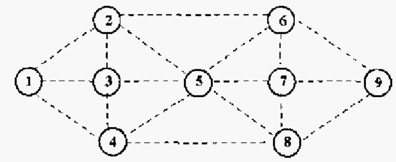


Fig. 6. An example wireless ad hoc network.

Figure 4 is such an algorithm for this purpose. In Figure 4, 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. Under such a lower bound, the corresponding physical paths are not necessarily feasible. In ALG-LB, b^* can be found using, e.g., the algorithm in [20] 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 $O(|E| \cdot \log |V|)$.

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

A proof of Proposition 1 is available in [19] and is omitted here for brevity. Note that 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. Also, $D(x_l^*)$ 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 estimation of D, it may not yield a pair of feasible paths for the s–t video session. In this section, we present a solution procedure that always produces a pair of feasible and near-optimal paths.

We believe that the best strategy to address Problem OPT-MM is to view it as a "black-box" optimization problem and explore an effective *metaheuristic* approach [10]. In particular, we find that *Genetic Algorithms* (GA) [9] are eminently suitable for addressing this type of complex combinatorial problems, most of which are multimodal and non-convex.[1] 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.

Figure 5 displays the flow chart for our GA-based approach to the MD multipath routing problem, which includes the following components: *solution representation, initialization, evaluation, selection, crossover,* and *mutation.* The termination condition in Figure 5 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 obtained in Section III. In what follows, we use an example ad hoc network shown in Figure 6 to illustrate the components in our GA-based approach.

1) Solution Representation and Initialization: In order to *encode* a feasible solution in the genetic format, we need to define a gene first and then map a solution to a sequence of genes (i.e., a chromosome). Such encoding should be suitable for fitness computation (which is determined by the objective function) and genetic operations. 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 [21]. For Problem OPT-MM, each feasible solution consists of a pair of paths (i.e., a pair of chromosomes), denoted as $[\mathcal{P}_1, \mathcal{P}_2]$. An individual in this case could be a pair of vectors containing the nodes on paths $\mathcal{P}_1$ and $\mathcal{P}_2$ (see, e.g., Figure 7).

Before entering the main loop in Figure 5, 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-

[1]The other two *evolutionary computation* (EC) metaheuristics, i.e., *evolutionary programming* (EP) and *evolutionary strategy*, are usually used for continuous optimization problems, while GA is mainly applied to combinatorial optimization problems [10].

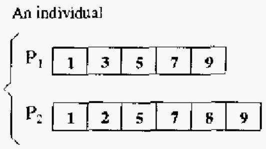


Fig. 7. An example individual, $s = 1$, $t = 9$.

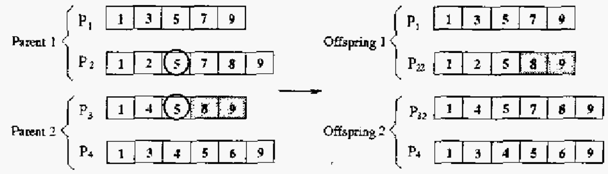


Fig. 8. An example of the crossover operation.

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, we select individuals that have a better chance or potential to produce "good" offsprings 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. Several selection schemes can be employed during this operation. For example, one possible scheme (known as *Roulette wheel* selection [9]) is to select an individual based on a probability in proportion to its normalized fitness value, i.e., $Pr\{$choosing individual $i\} = f(x_i)/\sum_j f(x_j)$. Another possible scheme (known as *Tournament* selection [9]) 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 offsprings. Crossover may create new individuals, thus exposing the search process to a new area of the fitness landscape. The decision of whether or not to perform a crossover operation is determined by the *crossover rate* θ.

Figure 8 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 offsprings 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 offsprings $[\mathcal{P}_1, \mathcal{P}_3]$ 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. This is a significant advantage over trajectory methods. However, 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 *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 fix 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.,

$$\begin{cases} \mu_0 = \mu_{max} \\ \mu_k = \mu_{max} - \frac{k \cdot (\mu_{max} - \mu_{min})}{T_{max}}, \end{cases} \tag{13}$$

where T_{max} is the maximum number of generations. Our results show that varying the mutation rates over generations significantly improves the on-line performance of the GA-based routing scheme. In essence, such schedule of μ is similar to the cooling schedule used in SA. Such a *hybridized* GA yields better convergence performance than a pure GA.

Figure 9 illustrates a simple example of the mutation operation. In this example, we implement mutation as follows. First, we choose a path $\mathcal{P}_h$, $h = 1$ or 2, with equal probabilities. Then, we randomly pick an integer value k in the interval $[2, |\mathcal{P}_h| - 1]$, 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 use a random 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 $\mathcal{P}_h^u \cup \mathcal{P}_h^d$. For the example in Figure 9, we randomly choose node 5 (as g_k) on path $\mathcal{P}_1$, and reconstruct a new segment starting from node 5 to the destination node 9, i.e., $\{5, 6, 9\}$. The new path created by mutation, $\hat{\mathcal{P}}_1$, is

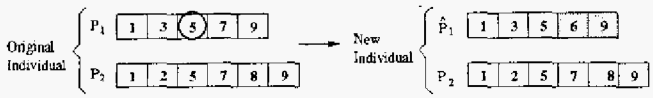


Fig. 9. An example of the mutation operation.

the union of the first half of the original path $\{1, 3, 5\}$ and the newly-constructed segment $\{5, 6, 9\}$. The new individual is $\hat{x} = [\hat{\mathcal{P}}_1, \mathcal{P}_2]$.

V. Numerical Results

In this section, we present numerical results on Problem OPT-MM. 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 source destination nodes s and t are randomly chosen. For every link, the failure probability is randomly chosen from [0.01, 0.3]; the available bandwidth is randomly chosen from [100, 400] Kb/s, with 50 Kb/s steps; the mean burst length is randomly chosen from [2,6]. 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 only affects the absolute value of distortion, but does not affect path selection decisions. The GA program 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 suggested solution to Problem OPT-MM.

A. GA-based Algorithm versus Exhaustive Search

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 deviation of the GA results is negligibly small, indicating that GA executions produce near-optimal or optimal solutions. 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.

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 tight in all of the cases, i.e., within 8% to 16% percentile of the global optimum.

TABLE II

Comparison of the average distortions obtained by the GA-based routing and exhaustive search

Topology	I	II	III	IV
Network Size	10-node	10-node	15-node	15-node
Global Optimal	0.3308	0.2004	0.3863	0.2969
GA (average)	0.3330	0.2004	0.3937	0.2972
GA (standard deviation)	7.6e-6	0	2.8e-5	2.9e-6
Lower Bound	0.2810	0.1832	0.3527	0.2444

B. Comparison with Trajectory Methods

In order to compare the GA-based approach to trajectory methods, we implemented simulated annealing (SA) and tabu search (TS), both of which have been used in solving certain networking problems. More information on our implementations of SA and TS can be found in [19] and is omitted here to conserve space.

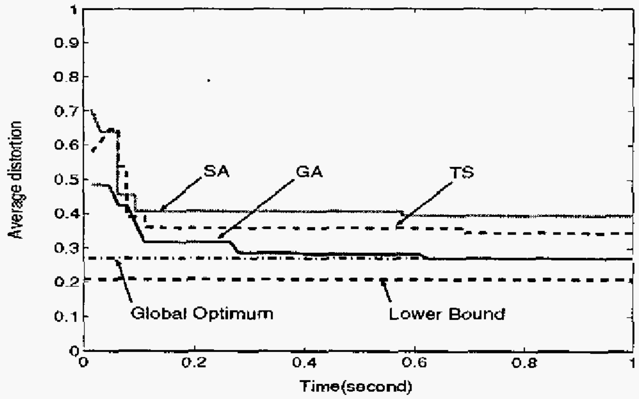
In Figure 10, we plot the evolution of distortion values obtained by GA, SA, and TS for a 10-node network and a 50-node network, respectively. All the three metaheuristics are terminated after running for 1 s. Upon termination, GA has evolved 210 generations in Figure 10(a) and 75 generations in Figure 10(b); SA ran for 1500 iterations in Figure 10(a) and 700 iterations in Figure 10(b); TS ran for 1050 iterations in Figure 10(a) and 550 generations in Figure 10(b). GA has fewer number of iterations than SA and TS, due to its higher computational complexity (see Section VI-B for more discussions). For both networks, the best distortion values found by GA are evidently much better than those by SA or TS. In Figure 10(a), GA converges to the global optimal very quickly, 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 Figure 10(b), although the global optimum is not obtainable here.

An interesting observation from Figure 10 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. Also note that the SA and TS curves increase at some time instances (e.g. the TS curve at 0.06 s in Figure 10(a) and the SA curve at 0.08 s in Figure 10(b)), which implies that a non-improving 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.

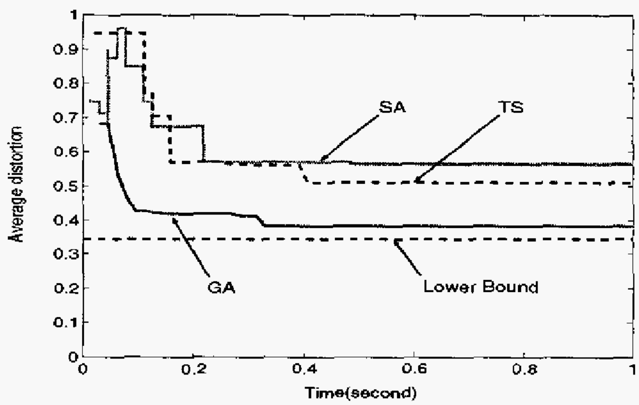
In addition to providing much better solutions, another strength of GA over trajectory methods is that multiple "good" solutions can be found after a single run. Such extra good paths can be used as alternative (or backup) paths if needed.

C. Comparison with Network-Centric Approaches

In this section, we compare our GA approach with network-centric routing approaches. In particular, we implement two popular network-centric multipath routing algorithms, namely



(a) Distortion evolution for a 10-node network.



(b) Distortion evolution for a 50-node network.

Fig. 10. Comparison of distortion evolution of three metaheuristic methods.

k-shortest path (SP) routing (with $k = 2$ or 2-SP) [22] and disjoint path routing, Disjoint Pathset Selection Protocol (DPSP) [23]. 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 real video clip.

There are many ways to generate MD video (see [1] for an excellent survey). We choose a time-domain partitioning coding scheme, where two descriptions are generated by separating the even and odd-numbered frames and coding them separately, as shown in Figure 11. The first frame in each stream is coded in the *intra-mode* (I frame), and the following frames are coded in the *inter-mode* (P frames). A 10% macroblock level intra-refreshment is used, which has been found to be effective in suppressing error propagation for the range of loss rates considered. This simple time-domain partitioning method is widely used in many video streaming studies [2], [4]–[7]. Compared with a traditional single description coder, this coder has a comparable computational complexity. Its coding efficiency is slightly lower than a single

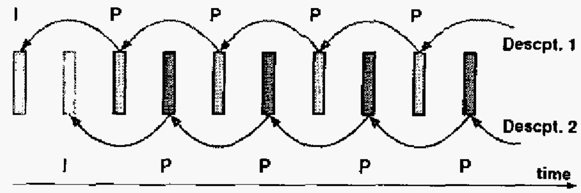


Fig. 11. The MD coding scheme used in the numerical examples.

TABLE III

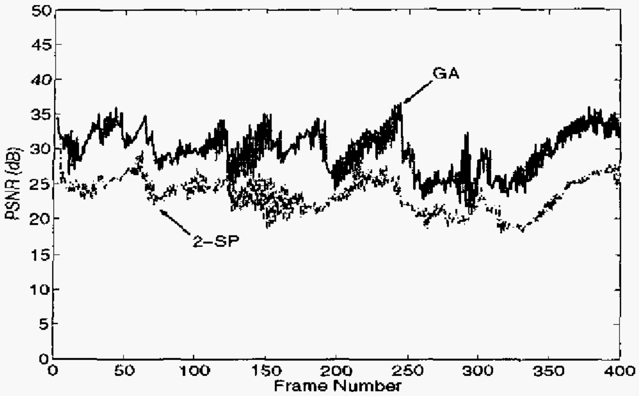
COMPARISON OF GA AND NETWORK-CENTRIC ROUTING

	$\mathcal{P}_1$ Loss Ratio	$\mathcal{P}_2$ Loss Ratio	Desc. 1 Bandwidth	Desc. 2 Bandwidth	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

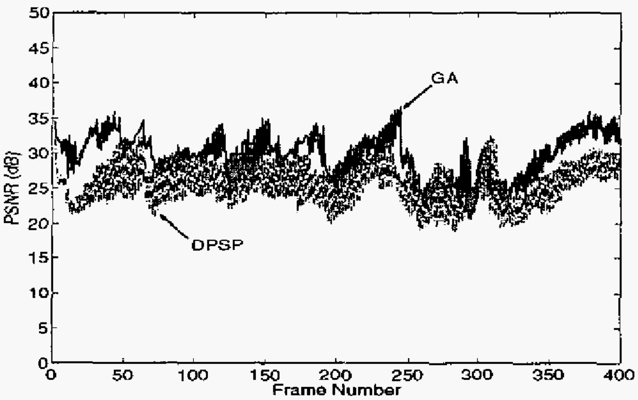
description coder, due to the fact that a longer motion prediction distance is used. However, this reduced coding efficiency is well justified by the resulting enhanced error resilience. The quarter common intermediate format (QCIF) [176 × 144 Y pixels/frame, 88×72 Cb/Cr pixels/frame] sequence "Foreman" (400 frames) is encoded at 15 fps for each description. 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 recently received frame that has not beed corrupted.

The quality of the paths found by the 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 Figure 12. 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 where link success probabilities are used in routing (as in DPSP). Even in this case, our GA-based routing still achieves a 1.27 dB improvement over this enhanced 2-SP version, which is still significant in terms of visual video quality. To further illustrate the significance of the improvements, we plot reconstructed Frame 235 in Figure 13. We observe that the image delivered by GA has the best quality, while the video frames delivered by 2-SP and DPSP are barely recognizable. The relatively low quality in these latter two frames are



(a) GA-based algorithm versus 2-SP.



(b) GA-based algorithm versus DPSP.

Fig. 12. PSNR curves of received video sequences.

caused by low encoding bit rates (which are determined by the bandwidths of the paths) and high packet loss rates (which are determined by the reliabilities of the paths), see (1) and (2). Obviously, our GA-based multipath routing offers significantly improved performance at the application layer over the two network-centric algorithms.

An inherent issue of transmitting MD 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 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) [4], or by striping packets of the descriptions across multiple paths to make losses of the descriptions relatively even.

(a) Original.

(b) GA-based routing.

(c) 2-SP.

(d) DPSP.

Fig. 13. Frame 235 from the original and decoded video sequences.

VI. Distributed Implementation

In addition to a centralized version of the GA-based multipath routing, we also investigate 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

Existing routing protocols can be roughly categorized as *proactive*, whereby a consistent and up-to-date view of the network is always maintained, and *reactive*, whereby route discovery is performed on-demand. We believe that the proposed GA-based routing is most suitable to be implemented within the pro-active 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 pro-active paradigm is well suited for this purpose, which can reduce session initiation delay for real-time multimedia applications. Second, for many applications (e.g., 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 (e.g., tracking each node's whereabouts).

Since an effective operation of cross-layer multipath routing requires the knowledge of a set of end-to-end paths, 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) [11] and *Topology Dissemination Based on Reverse-Path Forwarding* (TBRPF) [12], are very suitable for

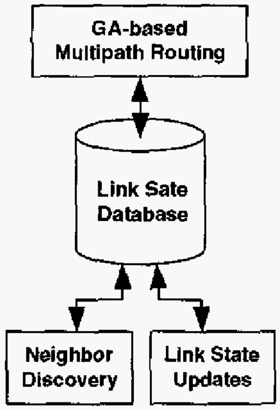


Fig. 14. distributed implementation architecture of the GA-based multipath routing.

this purpose. Figure 14 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 rely on any central entity in the network. A detailed discussion of the operations of its key modules can be found in [19].

B. Performance Issues

Link state routing protocols are quite successful in IETF standardization. In fact, two of the three existing RFCs for MANET routing, OLSR and TBRPF, follow the link state paradigm. It has been shown that the overhead associated with maintaining a link state database can be effectively minimized by either using MPRs, as in OLSR [11], or by reporting partial topology information in the LSAs and using "differential" HELLO messages that report only changes in neighbor status, as in TBRPF [12]. Our initial distributed implementation experience shows that, with proper design extension and changes, proactive distributed protocols can be used to build and maintain a link state database for the GA-based distributed implementation.

Another practical consideration for the GA-based approach is computation time. Since GA evolves a population of solutions over a large number of generations, it may have higher computational complexity than trajectory methods or network-centric multipath routing schemes. 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 Figure 10) in several hundred milliseconds, which are fast enough for practical uses. Incidentally, it has been shown in [21] that a GA-based algorithm is faster than Dijkstra's algorithm as the network size increases. 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 Figure 10). Therefore, there is a trade-off between solution optimality and computation time. For a delay-sensitive real-time application, GA can compute a set of "good" routes very quickly, and the application can use these "good" routes with a very small delay. As GA continues to evolve, the routes used by the application can be updated with newly computed routes for enhanced performance.

VII. Related Work

Multipath routing has been an active research area over the years. Various algorithms have been proposed to compute k-shortest paths [22], node- or link-disjoint paths [18], [23], or braided multiple paths [24]. For given multiple paths, traffic proportioning schemes are designed to disperse traffic to the paths for an increased end-to-end throughput, load balancing, and fast failure recovery [25]. In the area of wireless ad hoc networks, many existing routing protocols are multipath-capable (e.g., Terminode Routing [26]). 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 multimedia communications has recently been explored in [5], [7]. In [7], Begen *et al.* studied the multipath routing problem in the context of overlay networks. Although path selection is formulated as an optimization problem that minimizes video distortion, it is actually solved by an *exhaustive search* over the exponential solution space. The fast heuristic algorithm proposed in [8] can be used to speed up the computation by resorting to infrastructure support from the underlying network, which is not available in ad hoc networks. Several MD surrogate selection algorithms were presented in [5], 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 more important (and more difficult) problem of finding the optimal routes to the servers has not been addressed in this work.

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 [21], [27], [28], admission control [29], channel assignment [30], network design [31], scheduling [32] and buffer management [33]. In particular, Ahn and Ramakrishna [21] applied GA to the shortest path routing problem and compared its performance to Dijkstra's algorithm. The authors presented an elegant analysis on how to determine the optimal population size for the problem studied. 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 make a leap forward by exploring GA's potential to address the more complex *cross-layer* optimization problem. This problem is more substantial than the network-centric GA problems, since it not only requires knowledge at the network layer, but also, a deep understanding at the application layer (i.e., video coding capability) in order to fully exploit the design and optimization space across the layers.

VIII. Conclusions

In this paper, we studied the important problem of optimal multipath routing for MD video. We formulated the multipath routing problem following an application-centric cross-layer approach. 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 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 conditions. Further, we showed that a GA-based routing is amenable for distributed implementation in ad hoc routing protocols. This work provides an important methodology for addressing complex cross-layer optimization problems, particularly those involving the application and network layers.

Acknowledgments

This research has been 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 Grant No. N00014-03-1-0521.

References

[1] V. Goyal, "Multiple description coding: compression meets the network," *IEEE Signal Processing Mag.*, vol. 18, pp. 74–93, Sept. 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. Select. 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, Sept. 2002.

[4] J. Apostolopoulos and S. Wee, "Unbalanced multiple description video communication using path diversity," in *Proc. IEEE ICIP*, pp. 966–969, Thessaloniki, Greece, Oct. 2001.

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

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

[7] A. Begen, Y. Altunbasak, and O. Ergun, "Multi-path selection for multiple description encoded video streaming," *EURASIP Signal Processing: Image Communication*, to appear.

[8] ——, "Fast heuristics for multi-path selection for multiple description encoded video streaming," in *Proc. IEEE ICME*, pp. 517–520, Baltimore, Maryland, July 2003.

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

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

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

[12] R. Ogier, F. Templin, and M. Lewis, "Topology dissemination based on reverse-path forwarding (TBRPF)," IETF RFC 3684, Feb. 2004.

[13] A. Kopke, A. Willig, and H. Karl, "Chaotic maps as parsimonious bit error models of wireless channels," in *Proc. IEEE INFOCOM*, pp. 513–523, San Francisco, CA, Mar. 2003.

[14] S. Mao, S. Kompella, Y. Hou, H. Sherali, and S. Midkiff, "Optimal routing for multiple concurrent video sessions in wireless ad hoc networks," to appear in *Proc. IEEE ICC*, Seoul, Korea, May 2005.

[15] E. Setton, Y. Liang, and B. Girod, "Adaptive multiple description video streaming over multiple channels with active probing," in *Proc. IEEE ICME*, pp. I-509–I-512, Baltimore, MD, July 2003.

[16] 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.

[17] 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.

[18] 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.

[19] S. Mao, X. Lin, Y. Hou, H. Sherali, and S. Midkifff, "Multi-path routing for multiple description video over wireless ad hoc networks," Tech. Rep. The Bradley Dept. of ECE, Virginia Tech, Blacksburg, VA, Apr. 2004.

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

[21] 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.

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

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

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

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

[26] L. Blazevic, S. Giordano, and J. L. Boudec, "Self organized terminode routing," *J. of Cluster Comput.*, vol. 5, no. 2, pp. 205–218, Apr. 2002.

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

[28] 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.

[29] 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.

[30] 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, July 2003.

[31] 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.

[32] 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, Sept. 2003.

[33] 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.